CERTIFICATION EXAMINATIONS
FOR OKLAHOMA EDUCATORS™ (CEOE™)

OKLAHOMA SUBJECT AREA TESTS™ (OSAT™)

FIELD 081: COMPUTER SCIENCE

TEST FRAMEWORK

September 2008

	Subarea	Range of Competencies
I.	Computer Use in Educational Environments	01–02
II.	Computer System Concepts	03–06
III.	Program Design and Algorithms	07–10
IV.	Programming and Program Testing	11–14

OKLAHOMA SUBJECT AREA TESTS™ (OSAT™)

FIELD 081: COMPUTER SCIENCE
TEST FRAMEWORK

I. Computer Use in Educational Environments
II. Computer System Concepts
III. Program Design and Algorithms
IV. Programming and Program Testing

SUBAREA I—COMPUTER USE IN EDUCATIONAL ENVIRONMENTS

Competency 0001

Understand basic concepts related to the operation and use of computers and technology in educational environments.

The following topics are examples of content that may be covered under this competency.

- Demonstrate knowledge of procedures for installing and using peripherals and other devices and for troubleshooting hardware and software problems.
- Demonstrate knowledge of concepts and terminology related to telecommunications.
- Recognize types, characteristics, and uses of telecommunications tools and resources, including video conferencing and distance learning.
- Demonstrate familiarity with equity issues regarding computer use (e.g., students with special needs, students with limited English proficiency).
- Demonstrate familiarity with issues related to the privacy of electronic student information.

Competency 0002

Understand the characteristics and uses of productivity software.

The following topics are examples of content that may be covered under this competency.

- Demonstrate knowledge of the features and uses of word-processing and desktop-publishing software.
- Demonstrate knowledge of how to use spreadsheets to organize, analyze, and display data and how to integrate spreadsheet data into word-processing documents.
- Demonstrate knowledge of how to design and manipulate databases, including relational databases, to organize data and create custom reports.
- Demonstrate knowledge of terminology and concepts related to video and digital images (e.g., resolution, file formats, compression).
- Demonstrate knowledge of features and uses of video editing and graphic design software.
- Demonstrate familiarity with Web-page creation tools and concepts related to publishing material for the Web.

SUBAREA II—COMPUTER SYSTEM CONCEPTS

Competency 0003

Understand basic terminology related to computer architecture and characteristics of computer architecture.

The following topics are examples of content that may be covered under this competency.

- Demonstrate knowledge of terminology related to computer architecture.
- Identify characteristics and functions of computer components (e.g., CPU, RAM, secondary memory).
- Demonstrate knowledge of data representation at the machine level (e.g., floating point, integer, character).
- Demonstrate knowledge of data storage and how data is transferred from one location to another (e.g., registers, memory hierarchy).
- Recognize the steps in the machine cycle and their synchronization.
- Translate between binary, decimal, and hexadecimal number systems.

Competency 0004

Understand characteristics and functions of operating systems.

The following topics are examples of content that may be covered under this competency.

- Recognize the roles and functions of a computer's operating system.
- Compare the characteristics and functions of single-user and multiuser systems.
- Recognize the characteristics and functions of computer components and processes (e.g., devices, virtual memory, files, multitasking) and how the operating system manages them.
- Recognize characteristics and functions of utility programs.

Competency 0005

Understand types and characteristics of computer networks.

The following topics are examples of content that may be covered under this competency.

- Demonstrate knowledge of terminology and concepts related to computer networks.
- Demonstrate knowledge of various network configurations (e.g., peer to peer, client server) and their characteristics and the functions and characteristics of hubs, routers, and switches.
- Recognize characteristics of local area networks (LANs), wide area networks (WANs), and wireless configurations.
- Identify the basic structure and features of the Internet.
- Demonstrate knowledge of network protocols and concepts of data transfer on the Internet.
- Demonstrate knowledge of security issues related to networks and the Internet (e.g., firewalls, data encryption, malware, phishing).

Competency 0006

Understand the interactions between people and information systems and the social aspects of computing.

The following topics are examples of content that may be covered under this competency.

- Recognize the roles, responsibilities, and levels of access of various individuals who interact with information systems (e.g., programmer, Web site administrator, database administrator, end user).
- Analyze issues related to the legal and ethical use of computer technology, including security policies.
- Demonstrate knowledge of issues related to privacy and intellectual property rights when dealing with electronic data and information.
- Analyze significant historical events and trends related to computing.

SUBAREA III—PROGRAM DESIGN AND ALGORITHMS

Competency 0007

Understand principles and procedures for designing a program.

The following topics are examples of content that may be covered under this competency.

- Apply principles and concepts related to object-oriented programming.
- Demonstrate knowledge of the steps in the process of program design.
- Analyze flowcharts, schematic drawings, and pseudocode.
- Demonstrate knowledge of characteristics and uses of top-down, bottom-up, and object-oriented design methodologies.
- Apply principles of user interface design.

Competency 0008

Understand concepts and principles of modularization and data encapsulation in computer programming.

The following topics are examples of content that may be covered under this competency.

- Demonstrate knowledge of characteristics of program modules (e.g., functions, objects) and modularization strategies.
- Apply principles of inheritance, polymorphism, and abstraction in program design.
- Demonstrate knowledge of data encapsulation and its role in maintaining data integrity.
- Recognize characteristics and uses of libraries and predefined classes.
- Demonstrate knowledge of function calls, parameters, and parameter-passing techniques.
- Apply knowledge of the use of constructors in creating objects.
- Recognize characteristics of event-driven programming (e.g., input and output procedures, error handling).

Competency 0009

Understand characteristics and uses of algorithms in high-level languages.

The following topics are examples of content that may be covered under this competency.

- Understand the general characteristics and the role of algorithms in computing.
- Apply knowledge of the characteristics and uses of search algorithms (e.g., linear, binary).
- Demonstrate knowledge of the characteristics and uses of sorting algorithms.
- Determine algorithm output.
- Analyze various types of algorithms (e.g., time-and-space trade-offs, big-O notation).

Competency 0010

Understand principles and procedures for program development and implementation.

The following topics are examples of content that may be covered under this competency.

- Demonstrate knowledge of the steps in the programming process.
- Apply knowledge of how to develop robust programs with effective user interfaces.
- Recognize the purposes of programming style conventions (e.g., indenting, spacing, comments) and their appropriate application.
- Apply knowledge of object-oriented development strategies.
- Apply knowledge of strategies for modifying existing programs.
- Demonstrate knowledge of software tools for developing programs.
- Understand steps in the process of program execution (e.g., translation, linking, loading) in platform-dependent and platform-independent programming languages.

SUBAREA IV—PROGRAMMING AND PROGRAM TESTING

Competency 0011

Understand types and characteristics of programming languages.

The following topics are examples of content that may be covered under this competency.

- Recognize differences in levels of programming language (e.g., machine, assembly, high-level).
- Demonstrate knowledge of the characteristics, uses, strengths, and limitations of various types of high-level languages.
- Recognize characteristics and functions of compilers and interpreters.
- Demonstrate knowledge of the characteristics of various programming paradigms (e.g., imperative, functional, object-oriented).
- Demonstrate knowledge of the historical development of programming languages.

Competency 0012

Understand types and characteristics of statements, operators, and control structures in high-level languages.

The following topics are examples of content that may be covered under this competency.

- Recognize the characteristics and uses of operators.
- Recognize the characteristics and uses of statements (e.g., assignment, input/output, declaration).
- Demonstrate knowledge of the characteristics and uses of conditional control structures.
- Demonstrate knowledge of the characteristics and uses of repetitive control structures.

Competency 0013

Understand characteristics and applications of data types, structures, and abstraction mechanisms in high-level languages.

The following topics are examples of content that may be covered under this competency.

- Recognize the properties and uses of data types (e.g., integer, character, Boolean).
- Demonstrate knowledge of the characteristics and uses of constants, variables, classes, functions, and parameters.
- Analyze the characteristics and uses of inheritance and classes in object-oriented programming.
- Analyze the characteristics and uses of data structures (e.g., stacks, trees, arrays, heaps, linked lists).

Competency 0014

Understand basic concepts related to code analysis, program testing, and documentation.

The following topics are examples of content that may be covered under this competency.

- Demonstrate knowledge of program correctness issues and practices (e.g., testing program results, test data design).
- Analyze code segments to identify errors and determine output.
- Recognize common programming errors.
- Apply procedures for locating errors in computer programs.
- Recognize the characteristics and purposes of user and system documentation of programs.
- Apply principles of appropriate program documentation.
- Demonstrate knowledge of the use of comments.