# *COMPUTER SCIENCE STANDARDS*


This document was prepared by:

Office of Career, Technical and Adult Education
Nevada Department of Education
755 N. Roop Street, Suite 201
Carson City, NV 89701

Adopted by the State Board of Education /
State Board for Career and Technical Education on
October 4, 2013

## NEVADA STATE BOARD OF EDUCATION
## NEVADA STATE BOARD FOR CAREER AND TECHNICAL EDUCATION



### *CTE MISSION STATEMENT:*

The Office of Career, Technical and Adult Education is dedicated to developing innovative educational opportunities for students to acquire skills for productive employment and lifelong learning.

## NEVADA DEPARTMENT OF EDUCATION



# TABLE OF CONTENTS

## ACKNOWLEDGEMENTS

The development of Nevada career and technical standards and assessments is a collaborative effort sponsored by the Office of Career, Technical and Adult Education at the Department of Education and the Career and Technical Education Consortium of States. The Department of Education relies on teachers and industry representatives who have the technical expertise and teaching experience to develop standards and performance indicators that truly measure student skill attainment. Most important, however, is recognition of the time, expertise and great diligence provided by the writing team members in developing the career and technical standards for Computer Science.

## STANDARDS DEVELOPMENT MEMBERS

Roger Mayo, Instructor
Advanced Technologies Academy, Las Vegas

Jon Paul Ward, Instructor
Advanced Technologies Academy, Las Vegas

Lee Misch, Associate Professor
University of Nevada, Las Vegas

Kevin Mess
College of Southern Nevada, Las Vegas

Joel Fredrickson
Truckee Meadows Community College, Reno

## BUSINESS AND INDUSTRY VALIDATION

All CTE standards developed through the Nevada Department of Education are validated by business and industry through one or more of the following processes: (1) the standards are developed by a team consisting of business and industry representatives; or (2) a separate review panel was coordinated with industry experts to ensure the standards include the proper content; or (3) the adoption of nationally-recognized standards endorsed by business and industry.

The Computer Science standards were validated with the adoption of the nationally recognized standards approved the Computer Science Teachers Association and course description of the College Board's Advanced Placement Computer Science A course.

## PROJECT COORDINATOR

Melissa Scott, Education Programs Professional
Information and Media Technologies
Office of Career, Technical and Adult Education
Nevada Department of Education

## INTRODUCTION

The standards in this document are designed to clearly state what the student should know and be able to do upon completion of an advanced high school Computer Science program. These standards are designed for a three-credit course sequence that prepares the student for a technical assessment directly aligned to the standards.

These exit-level standards are designed for the student to complete all standards through their completion of a program of study. These standards are intended to guide curriculum objectives for a program of study.

The standards are organized as follows:

**Content Standards** are general statements that identify major areas of knowledge, understanding, and the skills students are expected to learn in key subject and career areas by the end of the program.

**Performance Standards** follow each content standard. Performance standards identify the more specific components of each content standard and define the expected abilities of students within each content standard.

**Performance Indicators** are very specific criteria statements for determining whether a student meets the performance standard. Performance indicators may also be used as learning outcomes, which teachers can identify as they plan their program learning objectives.

The crosswalk and alignment section of the document shows where the performance indicators support the English Language Arts and the Mathematics Common Core State Standards, and the Nevada State Science Standards. Where correlation with an academic standard exists, students in the Computer Science program perform learning activities that support, either directly or indirectly, achievement of one or more Common Core State Standards.

All students are encouraged to participate in the career and technical student organization (CTSO) that relates to their program area. CTSOs are co-curricular national associations that directly enforce learning in the CTE classroom through curriculum resources, competitive events, and leadership development. CTSOs provide students the ability to apply academic and technical knowledge, develop communication and teamwork skills, and cultivate leadership skills to ensure college and career readiness.

The Employability Skills for Career Readiness identify the “soft skills” needed to be successful in all careers, and must be taught as an integrated component of all CTE course sequences. These standards are available in a separate document.

The **Standards Reference Code** is only used to identify or align performance indicators listed in the standards to daily lesson plans, curriculum documents, or national standards.

| Program Name | Standards Reference Code |
|---|---|
| Computer Science | CS |

Example: CS.2.3.4

| Standards | Content Standard | Performance Standard | Performance Indicator |
|---|---|---|---|
| Computer Science | 2 | 3 | 4 |

| CONTENT STANDARD 1.0 : UNDERSTAND THE RELATIONSHIP BETWEEN HARDWARE AND SOFTWARE | |
|---|---|
| **PERFORMANCE STANDARD 1.1 : DEMONSTRATE KNOWLEDGE OF THE RELATIONSHIP BETWEEN HARDWARE AND SOFTWARE** | |
| 1.1.1 | Demonstrate proper use of industry-standard terminology |
| 1.1.2 | Examine the numbers systems: binary and hexadecimal |
| 1.1.3 | Describe machine limitations of finite representations (e.g., integer bounds, imprecision of floating-point representations, and round-off error) |
| 1.1.4 | Describe the central processing unit (CPU) and memory |
| 1.1.5 | Compare and contrast low- and high-level programming languages |

| CONTENT STANDARD 2.0 : UNDERSTAND CONCEPTS OF PROBLEM SOLVING AND ALGORITHM DEVELOPMENT | |
|---|---|
| **PERFORMANCE STANDARD 2.1 : UNDERSTAND A PROBLEM DESCRIPTION** | |
| 2.1.1 | Describe the problem |
| 2.1.2 | Demonstrate the solution(s) by hand |
| 2.1.3 | Explain how to validate the solution(s) |
| **PERFORMANCE STANDARD 2.2 : DEVELOP AN ALGORITHM** | |
| 2.2.1 | Identify expected input and output |
| 2.2.2 | Utilize basic steps in algorithmic problem solving |
| 2.2.3 | Use logical thinking to create an algorithm utilizing pseudo code and/or a flow chart |
| 2.2.4 | Discuss top-down versus bottom-up development |
| **PERFORMANCE STANDARD 2.3 : TEST ALGORITHMS** | |
| 2.3.1 | Generate test cases and expected results |
| 2.3.2 | Utilize test cases to walk through algorithms |
| 2.3.3 | Use results of the walk-through to adjust or modify algorithms |

| CONTENT STANDARD 3.0 : UNDERSTAND PROGRAMMING LANGUAGE CONCEPTS | |
|---|---|
| **PERFORMANCE STANDARD 3.1 : UTILIZE PROGRAMMING CONSTRUCTS** | |
| 3.1.1 | Differentiate between syntax and semantics |
| 3.1.2 | Incorporate primitive data types |
| 3.1.3 | Demonstrate input from different sources |
| 3.1.4 | Compare and contrast constants and variables |
| 3.1.5 | Select and implement conditional control |
| 3.1.6 | Select and implement iteration |
| 3.1.7 | Recognize and implement sequential control |
| 3.1.8 | Demonstrate output to different destinations |
| 3.1.9 | Design and implement user-defined data types |
| 3.1.10 | Select and implement recursion |
| 3.1.11 | Illustrate pointers and reference variables |
| **PERFORMANCE STANDARD 3.2 : PRACTICE PROCEDURAL PROGRAMMING** | |
| 3.2.1 | Design functions/methods |
| 3.2.2 | Properly apply scope (i.e., global versus local) |
| 3.2.3 | Select appropriate parameter passing by value or by reference |
| 3.2.4 | Select when to use void versus non-void functions/methods |
| **PERFORMANCE STANDARD 3.3 : PRACTICE OBJECT-ORIENTED PROGRAMMING (OOP)** | |
| 3.3.1 | Describe and implement abstract data types (i.e., data and functions) |
| 3.3.2 | Employ modularity and reusability |
| 3.3.3 | Employ encapsulation and information hiding |
| 3.3.4 | Select and implement composition (“has a”) and/or inheritance (“is a”) |
| 3.3.5 | Apply polymorphism |
| 3.3.6 | Establish abstract base classes and interfaces |

| CONTENT STANDARD 4.0 : DEVELOP PROGRAMS | |
|---|---|
| **PERFORMANCE STANDARD 4.1 : USE PROPER IMPLEMENTATION STRATEGIES** | |
| 4.1.1 | Evaluate and select language and tools (i.e., development environment, IDE, debugger) |
| 4.1.2 | Select procedural versus OOP paradigm |
| 4.1.3 | Distribute code among multiple files |
| 4.1.4 | Resolve runtime exceptions and handle errors |
| **PERFORMANCE STANDARD 4.2 : TEST AND DEBUG PROGRAMS** | |
| 4.2.1 | Employ debugging techniques (e.g., debugger, extra output statements, or hand-tracing codes) to identify and correct errors |
| 4.2.2 | Identify boundary cases and generate appropriate test data |
| 4.2.3 | Categorize errors (e.g., compile-time, run-time, logic, etc.) |
| 4.2.4 | Test classes and libraries in isolation |
| 4.2.5 | Perform integration testing of modules from multiple programmers |
| **PERFORMANCE STANDARD 4.3 : ANALYZE ALGORITHMS** | |
| 4.3.1 | Informally compare and contrast run times (i.e., best- and worst-case scenarios) |
| 4.3.2 | Assess algorithms using Big-O notation |

| CONTENT STANDARD 5.0 : CULTIVATE GOOD PROGRAMMING STYLE | |
|---|---|
| **PERFORMANCE STANDARD 5.1 : EMPLOY CODING STANDARDS** | |
| 5.1.1 | Use consistent naming conventions |
| 5.1.2 | Use meaningful identifiers |
| 5.1.3 | Use white space appropriately (e.g., indentation, blank lines, etc.) |
| 5.1.4 | Use named constants appropriately |
| 5.1.5 | Implement functions/methods to perform a single task |
| 5.1.6 | Produce a code that is clear, concise, and easy to maintain |
| 5.1.7 | Produce a code that compiles cleanly with no warning |
| **PERFORMANCE STANDARD 5.2 : DEMONSTRATE GOOD DOCUMENTATION SKILLS** | |
| 5.2.1 | Compose pre- and post-conditions for all functions and methods |
| 5.2.2 | Compose meaningful comments explaining critical or complex code |
| 5.2.3 | Describe program modules, variables, constants, and data types |
| 5.2.4 | Construct Unified Modeling Language (UML) class diagrams |

| CONTENT STANDARD 6.0 : UNDERSTAND STANDARD DATA STRUCTURES | |
|---|---|
| **PERFORMANCE STANDARD 6.1 : UTILIZE SIMPLE DATA TYPES** | |
| 6.1.1 | Select appropriate primitive types |
| 6.1.2 | Implement declarations and initialization |
| 6.1.3 | Explain operators and order of operations |
| **PERFORMANCE STANDARD 6.2 : DEMONSTRATE KNOWLEDGE OF ARRAYS** | |
| 6.2.1 | Manipulate strings as arrays |
| 6.2.2 | Assess implementation strategy (i.e., static or dynamic) |
| 6.2.3 | Access arrays (i.e., sequential, random) |
| 6.2.4 | Search arrays (i.e., sequential, binary) |
| 6.2.5 | Sort arrays (i.e., bubble, selection, insertion, and merge) |
| **PERFORMANCE STANDARD 6.3 : DEMONSTRATE KNOWLEDGE OF CLASSES** | |
| 6.3.1 | Compare and contrast classes and objects |
| 6.3.2 | Implement class declaration |
| 6.3.3 | Create constructors and destructors |
| 6.3.4 | Implement overloaded and overridden functions/methods |
| 6.3.5 | Apply OOP concepts, including inheritance, polymorphism, interfaces, and abstract classes |
| 6.3.6 | Manipulate strings as objects |
| 6.3.7 | Explore template/generics |
| **PERFORMANCE STANDARD 6.4 : DEMONSTRATE KNOWLEDGE OF LISTS** | |
| 6.4.1 | Use singly-linked lists |
| 6.4.2 | Explore doubly- and circularly-linked lists |
| 6.4.3 | Assess implementation strategy (i.e., array, dynamic) |
| 6.4.4 | Access list with and without iterators |
| 6.4.5 | Search lists (i.e., sequential and binary) |
| 6.4.6 | Implement insertion and deletion functions/methods |
| 6.4.7 | Compare and contrast sorting algorithms (i.e., bubble, selection, insertion, and merge) |
| 6.4.8 | Select and implement stacks and queues |
| 6.4.9 | Use iterators |

| CONTENT STANDARD 7.0 : DEMONSTRATE KNOWLEDGE OF COMPUTING IN SOCIETAL CONTEXT | |
|---|---|
| **PERFORMANCE STANDARD 7.1 : RECOGNIZE THE SYSTEM RELIABILITY ISSUES** | |
| 7.1.1 | Research examples of system failures and their impact |
| 7.1.2 | Explain cross-platform issues |
| **PERFORMANCE STANDARD 7.2 : EXAMINE ETHICAL AND LEGAL ISSUES** | |
| 7.2.1 | Debate intellectual property, patent, and copyright laws |
| 7.2.2 | Describe the meaning of privacy in relation to the use of technology |
| 7.2.3 | Describe conflict of interest |
| 7.2.4 | Analyze non-disclosure agreements |
| 7.2.5 | Compare and contrast the code of ethics of the Institute of Electrical and Electronic Engineers (IEEE) and Association for Computing Machinery (ACM) |

This Page was Intentionally Left Blank

# CROSSWALKS AND ALIGNMENTS OF COMPUTER SCIENCE STANDARDS AND THE COMMON CORE STATE STANDARDS, THE NEVADA SCIENCE STANDARDS, AND THE COMMON CAREER TECHNICAL CORE STANDARDS

## CROSSWALKS (ACADEMIC STANDARDS)

The crosswalk of the Computer Science Standards shows links to the Common Core State Standards for English Language Arts and Mathematics and the Nevada Science Standards. The crosswalk identifies the performance indicators in which the learning objectives in the Computer Science program support academic learning. The performance indicators are grouped according to their content standard and are crosswalked to the English Language Arts and Mathematics Common Core State Standards and the Nevada Science Standards.

## ALIGNMENTS (MATHEMATICAL PRACTICES)

In addition to correlation with the Common Core Mathematics Content Standards, many performance indicators support the Common Core Mathematical Practices. The following table illustrates the alignment of the Computer Science Standards Performance Indicators and the Common Core Mathematical Practices. This alignment identifies the performance indicators in which the learning objectives in the Computer Science program support academic learning.

## CROSSWALKS (COMMON CAREER TECHNICAL CORE)

The crosswalk of the Computer Science Standards shows links to the Common Career Technical Core. The crosswalk identifies the performance indicators in which the learning objectives in the Computer Science program support the Common Career Technical Core. The Common Career Technical Core defines what students should know and be able to do after completing instruction in a program of study. The Computer Science Standards are crosswalked to the Information Technology Career Cluster™ (IT) and the Programming & Software Development Career Pathway (IT-PRG) Career Pathway.

This Page was Intentionally Left Blank

# CROSSWALK OF COMPUTER SCIENCE STANDARDS AND THE COMMON CORE STATE STANDARDS

## CONTENT STANDARD 1.0: UNDERSTAND THE RELATIONSHIP BETWEEN HARDWARE AND SOFTWARE

| Performance Indicators | Common Core State Standards and Nevada Science Standards |
|---|---|
| 1.1.1 | **English Language Arts: Writing Standards for Literacy in Science and Technical Subjects**<br>WHST.11-12.4 Produce clear and coherent writing in which the development, organization, and style are appropriate to task, purpose, and audience. |
| 1.3.1 | **Math: Number & Quantity – The Real Number System**<br>NRN.A.2 Rewrite expressions involving radicals and rational exponents using the properties of exponents.<br>**Math: Number & Quantity – Quantities**<br>NQ.A.3 Choose a level of accuracy appropriate to limitations on measurement when reporting quantities. |

## CONTENT STANDARD 2.0: UNDERSTAND CONCEPTS OF PROBLEM SOLVING AND ALGORITHM DEVELOPMENT

| Performance Indicators | Common Core State Standards and Nevada Science Standards |
|---|---|
| 2.2.2 | **Math: Algebra – Reasoning with Equations and Inequalities**<br>AREI.A.1 Explain each step in solving a simple equation as following from the equality of numbers asserted at the previous step, starting from the assumption that the original equation has a solution. Construct a viable argument to justify a solution method. |

## CONTENT STANDARD 3.0: UNDERSTAND PROGRAMMING LANGUAGE CONCEPTS

| Performance Indicators | Common Core State Standards and Nevada Science Standards | |
|---|---|---|
| 3.2.1 | **Math: Functions – Interpreting Functions** | |
| | FIF.A.2 | Use function notations, evaluate functions for inputs in their domains, and interpret statements that use function notation in terms of a context. |
| | FIF.A.3 | Recognize that sequences are functions, sometimes defined recursively, whose domain is a subset of the integers. |

## CONTENT STANDARD 5.0: CULTIVATE GOOD PROGRAMMING STYLE

| Performance Indicators | Common Core State Standards and Nevada Science Standards |
|---|---|
| 5.1.1 | **English Language Arts: Writing Standards for Literacy in Science and Technical Subjects**<br>WHST.11-12.2d Use precise language, domain-specific vocabulary and techniques such as metaphor, simile, and analogy to manage the complexity of the topic; convey a knowledgeable stance in a style that responds to the discipline and context as well as to the expertise of likely readers. |
| 5.1.5 | **Math: Algebra – Creating Equations**<br>ACED.A.1 Create equations and inequalities in one variable and use them to solve problems. |
| 5.1.6 | **English Language Arts: Writing Standards for Literacy in Science and Technical Subjects**<br>WHST.11-12.2 Write informative/explanatory texts, including the narration of historical events, scientific procedures/ experiments, or technical processes. |
| 5.2.1 | **English Language Arts: Writing Standards for Literacy in Science and Technical Subjects**<br>WHST.11-12.8 Gather relevant information from multiple authoritative print and digital sources, using advanced searches effectively; assess the strengths and limitations of each source in terms of the specific task, purpose, and audience; integrate information into the text selectively to maintain the flow of ideas, avoiding plagiarism and overreliance on any one source and following a standard format for citation. |
| 5.2.3 | **English Language Arts: Writing Standards for Literacy in Science and Technical Subjects**<br>WHST.11-12.4 Produce clear and coherent writing in which the development, organization, and style are appropriate to task, purpose, and audience. |

## CONTENT STANDARD 6.0: UNDERSTAND STANDARD DATA STRUCTURES

| Performance Indicators | Common Core State Standards and Nevada Science Standards |
|---|---|
| 6.1.3 | **English Language Arts: Writing Standards for Literacy in Science and Technical Subjects**<br>WHST.11-12.8 Gather relevant information from multiple authoritative print and digital sources, using advanced searches effectively; assess the strengths and limitations of each source in terms of the specific task, purpose, and audience; integrate information into the text selectively to maintain the flow of ideas, avoiding plagiarism and overreliance on any one source and following a standard format for citation. |
| 6.4.7 | **English Language Arts: Writing Standards for Literacy in Science and Technical Subjects**<br>WHST.11-12.8 Gather relevant information from multiple authoritative print and digital sources, using advanced searches effectively; assess the strengths and limitations of each source in terms of the specific task, purpose, and audience; integrate information into the text selectively to maintain the flow of ideas, avoiding plagiarism and overreliance on any one source and following a standard format for citation. |

## CONTENT STANDARD 7.0: DEMONSTRATE KNOWLEDGE OF COMPUTING IN SOCIETAL CONTEXT

| Performance Indicators | Common Core State Standards and Nevada Science Standards |
|---|---|
| 7.1.1 | **English Language Arts: Writing Standards for Literacy in Science and Technical Subjects**<br>WHST.11-12.7 Conduct short as well as more sustained research projects to answer a question (including a self-generated question) or solve a problem; narrow or broaden the inquiry when appropriate; synthesize multiple sources on the subject, demonstrating understanding of the subject under investigation. |
| 7.2.1 | **English Language Arts: Reading Standards for Literacy in Science and Technical Subjects**<br>RST.11-12.9 Synthesize information from a range of sources (e.g., texts, experiments, simulations) into a coherent understanding of a process, phenomenon, or concept, resolving conflicting information when possible.<br>**English Language Arts: Speaking and Listening Standards**<br>SL.11-12.4 Present information, findings, and supporting evidence, conveying a clear and distinct perspective, such that listeners can follow the line of reasoning, alternative or opposing perspectives are addressed, and the organization, development, substance, and style are appropriate to purpose, audience, and a range of formal and informal tasks. |
| 7.2.2 | **English Language Arts: Speaking and Listening Standards**<br>SL.11-12.1d Respond thoughtfully to diverse perspectives; synthesize comments, claims, and evidence made on all sides of an issue; resolve contradictions when possible; and determine what additional information or research is required to deepen the investigation or complete the task. |
| 7.2.3 | **English Language Arts: Speaking and Listening Standards**<br>SL.11-12.4 Present information, findings, and supporting evidence, conveying a clear and distinct perspective, such that listeners can follow the line of reasoning, alternative or opposing perspectives are addressed, and the organization, development, substance, and style are appropriate to purpose, audience, and a range of formal and informal tasks.<br>**English Language Arts: Writing Standards for Literacy in Science and Technical Subjects**<br>WHST.11-12.4 Produce clear and coherent writing in which the development, organization, and style are appropriate to task, purpose, and audience. |
| 7.2.4 | **English Language Arts: Writing Standards for Literacy in Science and Technical Subjects**<br>WHST.11-12.8 Gather relevant information from multiple authoritative print and digital sources, using advanced searches effectively; assess the strengths and limitations of each source in terms of the specific task, purpose, and audience; integrate information into the text selectively to maintain the flow of ideas, avoiding plagiarism and overreliance on any one source and following a standard format for citation. |
| 7.2.5 | **English Language Arts: Writing Standards for Literacy in Science and Technical Subjects**<br>WHST.11-12.8 Gather relevant information from multiple authoritative print and digital sources, using advanced searches effectively; assess the strengths and limitations of each source in terms of the specific task, purpose, and audience; integrate information into the text selectively to maintain the flow of ideas, avoiding plagiarism and overreliance on any one source and following a standard format for citation. |

## ALIGNMENT OF COMPUTER SCIENCE STANDARDS AND THE COMMON CORE MATHEMATICAL PRACTICES

| Common Core Mathematical Practices | Computer Science Performance Indicators |
|---|---|
| 1. Make sense of problems and persevere in solving them. | |
| 2. Reason abstractly and quantitatively. | |
| 3. Construct viable arguments and critique the reasoning of others. | |
| 4. Model with mathematics. | 2.1.3; 2.2.3<br>5.2.1 |
| 5. Use appropriate tools strategically. | |
| 6. Attend to precision. | |
| 7. Look for and make use of structure. | 6.1.4; 6.2.1-6.2.5 |
| 8. Look for and express regularity in repeated reasoning. | 3.1.10; 3.2.1, 3.2.3 |

## CROSSWALKS OF COMPUTER SCIENCE STANDARDS AND THE COMMON CAREER TECHNICAL CORE

| Information Technology Career Cluster™ (IT) | Performance Indicators |
|---|---|
| 1. Demonstrate effective professional communication skills and practices that enable positive customer relationships. | 1.1.1 |
| 2. Use product or service design processes and guidelines to produce a quality information technology (IT) product or service. | 2.1.1<br>4.1.1-4.1.4; 4.2.1-4.2.5 |
| 3. Demonstrate the use of cross-functional teams in achieving IT project goals. | |
| 4. Demonstrate positive cyber citizenry by applying industry accepted ethical practices and behaviors. | 7.2.1-7.2.5 |
| 5. Explain the implications of IT on business development. | 7.1.1-7.1.2 |
| 6. Describe trends in emerging and evolving computer technologies and their influence on IT practices. | |
| 7. Perform standard computer backup and restore procedures to protect IT information. | 1.3.1, 1.4.1 |
| 8. Recognize and analyze potential IT security threats to develop and maintain security requirements. | 7.1.1-7.1.2 |
| 9. Describe quality assurance practices and methods employed in producing and providing quality IT products and services. | 5.1.1-5.1.7 |
| 10. Describe the use of computer forensics to prevent and solve information technology crimes and security breaches. | |
| 11. Demonstrate knowledge of the hardware components associated with information systems. | 1.1.1 |
| 12. Compare key functions and applications of software and determine maintenance strategies for computer systems. | 1.1.1-1.1.5 |

| Programming & Software Development Career Pathway (IT-PRG) | Performance Indicators |
|---|---|
| 1. Analyze customer software needs and requirements. | 2.1.1-2.1.3 |
| 2. Demonstrate the use of industry standard strategies and project planning to meet customer specifications. | |
| 3. Analyze system and software requirements to ensure maximum operating efficiency. | 2.1.1-2.1.3<br>2.2.1 |
| 4. Demonstrate the effective use of software development tools to develop software applications. | 3.2.1-3.2.4<br>3.3.1-3.3.6 |
| 5. Apply an appropriate software development process to design a software application. | 2.2.1-2.2.4 |
| 6. Program a computer application using the appropriate programming language. | 3.1.1-3.1.11 |
| 7. Demonstrate software testing procedures to ensure quality products. | 2.3.1-2.3.3<br>4.2.1-4.2.5 |
| 8. Perform quality assurance tasks as part of the software development cycle. | 2.3.1-2.3.3 |
| 9. Perform software maintenance and customer support functions. | |
| 10. Design, create and maintain a database. | 6.1.1-6.1.3 |