TEXAS EDUCATOR CERTIFICATION

TExES | TExMaT | TASC/TASC-ASL

Texas Examinations of Educator Standards™ (TExES™) Program

Preparation Manual

Computer Science 8–12 (241)

Table of Contents

NOTE: After clicking on a link, right click and select "Previous View" to go back to original text.

About The Test

Test Name	Computer Science 8–12
Test Code	241
Time	5 hours
Number of Questions	100 selected-response questions
Format	Computer-administered test (CAT)

The TExES Computer Science 8–12 (241) test is designed to assess whether a test taker has the requisite knowledge and skills that an entry-level educator in this field in Texas public schools must possess. The 100 selected-response questions are based on the Computer Science 8–12 test framework. The test may contain questions that do not count toward the score. Your final scaled score will be based only on scored questions.

NOTE: After clicking on a link, right click and select "Previous View" to go back to original text.

The Domains

Domain	Domain Title	Approx. Percentage of Test	Standards Assessed
I.	Technology Applications Core	12.5%	Computer Science 8–12: I–VII
II.	Program Design and Development	35%	Computer Science 8–12: VIII
III.	Programming Language Topics	40%	Computer Science 8–12: VIII
IV.	Specialized Topics	12.5%	Computer Science 8–12: VIII, IX, XI, XIV

NOTE: After clicking on a link, right click and select "Previous View" to go back to original text.

The Standards

Computer Science 8–12 Standard I
All teachers use and promote creative thinking and innovative processes to construct knowledge, generate new ideas and create products.

Computer Science 8–12 Standard II
All teachers collaborate and communicate both locally and globally using digital tools and resources to reinforce and promote learning.

Computer Science 8–12 Standard III
All teachers acquire, analyze and manage content from digital resources.

Computer Science 8–12 Standard IV
All teachers make informed decisions by applying critical-thinking and problem-solving skills.

Computer Science 8–12 Standard V
All teachers practice and promote safe, responsible, legal and ethical behavior while using technology tools and resources.

Computer Science 8–12 Standard VI
All teachers demonstrate a thorough understanding of technology concepts, systems and operations.

Computer Science 8–12 Standard VII
All teachers know how to plan, organize, deliver and evaluate instruction for all students that incorporates the effective use of current technology for teaching and integrating the Technology Applications Texas Essential Knowledge and Skills (TEKS) into the curriculum.

Computer Science 8–12 Standard VIII
The computer science teacher has the knowledge and skills needed to teach the creativity and innovation; communication and collaboration; research and information fluency; critical thinking, problem solving and decision making; digital citizenship; and technology operations and concepts strands of the Technology Applications Texas Essential Knowledge and Skills (TEKS) in computer science, in addition to the content described in Technology Applications Standards I–VII.

Computer Science 8–12 Standard IX
The digital forensics teacher has the knowledge and skills needed to teach the creativity and innovation; communication and collaboration; research and information fluency; critical thinking, problem solving and decision making; digital citizenship; and technology operations and concepts strands of the Technology Applications Texas Essential Knowledge and Skills (TEKS) in digital forensics, in addition to the content described in Technology Applications Standards I–VII.

NOTE: After clicking on a link, right click and select "Previous View" to go back to original text.

Computer Science 8–12 Standard XI
The robotics teacher has the knowledge and skills needed to teach the creativity and innovation; communication and collaboration; research and information fluency; critical thinking, problem solving and decision making; digital citizenship; and technology operations and concepts strands of the Technology Applications Texas Essential Knowledge and Skills (TEKS) in robotics, in addition to the content described in Technology Applications Standards I–VII.

Computer Science 8–12 Standard XIV
The game/application development teacher has the knowledge and skills needed to teach the creativity and innovation; communication and collaboration; research and information fluency; critical thinking, problem solving and decision making; digital citizenship; and technology operations and concepts strands of the Technology Applications Texas Essential Knowledge and Skills (TEKS) in game/application development, in addition to the content described in Technology Applications Standards I–VII.

NOTE: After clicking on a link, right click and select "Previous View" to go back to original text.

Domains and Competencies

The content covered by this test is organized into broad areas of content called **domains**. Each domain covers one or more of the educator standards for this field. Within each domain, the content is further defined by a set of **competencies**. Each competency is composed of two major parts:

- The **competency statement**, which broadly defines what an entry-level educator in this field in Texas public schools should know and be able to do.
- The **descriptive statements**, which describe in greater detail the knowledge and skills eligible for testing.

Domain I — Technology Applications Core

Competency 001: *The computer science teacher knows technology terminology and concepts; the appropriate use of hardware, software and digital files; and how to acquire, analyze and evaluate digital information.*

The beginning teacher:

A. Knows and uses technology terminology and concepts appropriate to the task.

B. Knows the appropriate use of software and hardware components.

C. Demonstrates knowledge of various types of networks (e.g., LAN, WAN, intranets and the Internet).

D. Knows how to select, connect and use a variety of local and remote peripheral devices.

E. Knows how to manage compatibility issues for a variety of media, file formats (e.g., text, graphics, image, video, audio), file naming conventions, file management structures and digital organization strategies.

F. Knows how to evaluate software for quality, appropriateness, effectiveness, efficiency, support and licensing to make decisions regarding its proper acquisition and use.

G. Knows how to access, manage and manipulate information from secondary storage devices.

H. Knows strategies for searching, acquiring and accessing information from electronic resources.

I. Knows how to assess the accuracy and validity of acquired information and how to resolve information conflicts through research and comparison of data from multiple sources.

J. Demonstrates knowledge of intellectual property rights (e.g., copyright, Creative Commons, free and open source licensing) when accessing, using, manipulating and editing electronic data.

NOTE: After clicking on a link, right click and select "Previous View" to go back to original text.

K. Demonstrates knowledge of issues of unacceptable use of computer resources including, but not limited to, cyberbullying and harassment, computer hacking, computer piracy, plagiarism, vandalism, intentional virus setting and invasion of privacy.

L. Demonstrates ethical and lawful acquisition of digital information, including the use of established methods to cite sources.

M. Understands digital safety, privacy rules, digital etiquette, acceptable use of technology and the ethical and legal responsibilities of using social media.

N. Knows how to use online help and other support documentation to troubleshoot minor technical problems with hardware and software.

O. Knows how to develop documentation for a variety of products.

P. Demonstrates knowledge of technology's historical and future impact on society.

Competency 002: *The computer science teacher knows how to use technology tools to solve problems, evaluate results and communicate information in a variety of formats for diverse audiences.*

The beginning teacher:

A. Knows how to plan, create, edit, analyze and represent data in documents using general productivity software.

B. Knows how to explore complex concepts using simulations, models, interactive virtual environments and new technologies to develop hypotheses, modify input and analyze results.

C. Demonstrates knowledge of how to design and implement procedures to track trends, set timelines and evaluate the progress of products using project management tools for continual improvement in process and product development.

D. Knows how to evaluate projects for design, purpose, audience and content delivery using various criteria (e.g., project specifications, rubrics).

E. Knows how to select representative products to be collected and stored in an electronic evaluation tool and to evaluate products for relevance to the assignment or task.

F. Knows how to plan and design products that are accessible to learners with diverse needs and abilities.

NOTE: After clicking on a link, right click and select "Previous View" to go back to original text.

Competency 003: *The computer science teacher knows how to plan, organize, deliver and evaluate instruction that effectively utilizes current technology for teaching the Technology Applications Texas Essential Knowledge and Skills (TEKS) to all students.*

The beginning teacher:

A. Knows how to implement developmentally appropriate instructional practices, activities and materials to improve student learning.

B. Knows how to implement lessons using diverse instructional strategies.

C. Demonstrates knowledge of issues related to the equitable use of technology for diverse populations.

D. Knows how to implement instruction that allows students to solve problems by posing questions, collecting data and interpreting results.

E. Knows how to develop and facilitate collaborative tasks among group members, incorporating diverse perspectives while exploring alternative solutions.

F. Knows strategies to help students learn how to locate, retrieve, analyze, evaluate, communicate and retain content-related information from a variety of texts and digital sources.

G. Knows how to evaluate student projects and portfolios using various assessment methods (e.g., formal, informal).

H. Knows how to promote effective self-evaluation and use of feedback from peers.

I. Knows the relationship between instruction and assessment.

J. Knows how to adjust instruction based on assessment results.

K. Demonstrates knowledge of emerging technology and its role in education.

L. Knows the importance of self-assessment and planning for professional growth.

Domain II — Program Design and Development

Competency 004: *The computer science teacher knows problem-solving strategies and different procedures for program design*.

The beginning teacher:

A. Exhibits knowledge of all phases of the software system life cycle and understands its cyclical nature.

B. Knows the characteristics of programming design strategies (e.g., design specification, top-down design, step-wise refinement, black box, object-oriented design).

NOTE: After clicking on a link, right click and select "Previous View" to go back to original text.

C. Knows how to apply problem-solving strategies to implement design.

D. Demonstrates the use of visual organizers (e.g., flow diagrams, Unified Modeling Language [UML]) to document program designs and implementations.

E. Knows how to create robust programs with emphasis on design to facilitate maintenance, program expansion, reliability, validity and efficiency.

Competency 005: *The computer science teacher knows procedures for software development and implementation.*

The beginning teacher:

A. Knows the characteristics of models used in the development of software systems.

B. Demonstrates the ability to use an integrated development environment (IDE).

C. Demonstrates knowledge of collaborative strategies for the development of complex software systems (e.g., design/implementation teams, software validation/testing, risk assessment).

D. Demonstrates the ability to work independently or collaboratively to implement a solution to a problem according to design specifications, identifying data types, objects needed, subtasks to be performed and reusable components from existing code.

E. Demonstrates the use of programming style conventions (e.g., spacing, indentation, descriptive identifiers, comments, documentation, standardized programming style) to enhance the readability and functionality of code.

F. Knows how to create robust programs with emphasis on programming style to facilitate maintenance, program expansion, reliability, validity and efficiency.

G. Knows how to create and use libraries of generic modular code for efficient programming.

H. Demonstrates the ability to read, evaluate, correct and improve existing code.

I. Knows how to create robust programs by avoiding runtime errors and handling anticipated errors (e.g., correct handling of input and output, division by zero, type mismatch).

J. Demonstrates the ability to test programs by entering valid and invalid data; investigating boundary conditions; testing classes, methods and libraries in isolation; and performing stepwise refinement.

K. Demonstrates the ability to debug program errors (e.g., syntax, runtime, logic) using error messages, reference materials, language documentation and other effective strategies.

Competency 006: *The computer science teacher knows computer science terminology and concepts and the characteristics of different programming languages and paradigms.*

The beginning teacher:

A. Knows fundamental computer science vocabulary, including terms related to hardware, software and computational thinking.

B. Knows specific programming terminology, including terms related to data type, data structures, algorithms and programming constructs.

C. Knows the differences between low-level and high-level languages.

D. Knows the differences between compiled and interpreted languages.

E. Knows the characteristics of and differences in current programming languages and paradigms (e.g., procedural, object-oriented).

Domain III — Programming Language Topics

Competency 007: *The computer science teacher correctly and efficiently uses data types, data structures and functions in the development of code.*

The beginning teacher:

A. Understands various computer-related number-base systems and uses them to count, convert and perform mathematical operations.

B. Understands the characteristics of standard data types in current programming languages (e.g., integer, floating point, character, string, Boolean).

C. Demonstrates the ability to develop code using constants, variables, data structures and appropriate scope (e.g., local, global).

D. Understands the difference between primitive and referenced data types (e.g., objects, lists).

E. Demonstrates the ability to cast between data types and provide object functionality to primitive data types.

F. Demonstrates effective use of standard libraries (e.g., math, string) in the development of code.

G. Demonstrates the ability to create user-defined functions and procedures.

H. Demonstrates understanding of the difference between parameters that are passed by value or by reference.

I. Knows how to identify object-oriented data types and how to delineate the advantages and disadvantages of object data.

NOTE: After clicking on a link, right click and select "Previous View" to go back to original text.

J. Demonstrates the ability to process data in one-dimensional and multi-dimensional arrays.

K. Understands how to implement input/output processes (e.g., file, keyboard).

L. Demonstrates the ability to manipulate text using string processing functions (e.g., concatenation, substring, search).

M. Understands concepts related to the traversal and processing of abstract data types (e.g., stacks, queues, linked lists, trees, graphs).

Competency 008: *The computer science teacher correctly and efficiently uses statements and control structures in the development of code.*

The beginning teacher:

A. Creates mathematical expressions using arithmetic operators (addition, subtraction, multiplication, division, integer division and modulus division).

B. Develops programs using standard operators (e.g., arithmetic, relational, logical, assignment), operator precedence and short-circuit evaluation.

C. Demonstrates an understanding of conditional and iterative control structures.

D. Demonstrates coding proficiency in current programming languages, including an object-oriented language.

E. Demonstrates understanding of object-oriented design and the relationships (including composition and inheritance) among defined classes, abstract classes and interfaces.

F. Designs classes that encapsulate data and related methods.

G. Demonstrates understanding of polymorphism in overloading and overriding features of classes.

H. Demonstrates the ability to use abstract classes and interfaces to design and implement multi-class programs.

Competency 009: *The computer science teacher knows how to construct, compare and analyze various algorithms.*

The beginning teacher:

A. Constructs searching algorithms (e.g., linear and binary searches).

B. Constructs sorting algorithms including, but not limited to, selection, insertion, merge and quick sorts.

C. Analyzes the best-, average- and worst-case run-time efficiencies of various algorithms using informal comparisons and Big-O notation.

D. Traces, compares and uses iterative and recursive algorithms.

NOTE: After clicking on a link, right click and select "Previous View" to go back to original text.

E. Demonstrates an understanding of and the ability to develop common algorithms to solve practical problems.

Domain IV — Specialized Topics

Competency 010: *The computer science teacher knows discrete mathematics topics relevant to computer science.*

The beginning teacher:

A. Demonstrates knowledge of terminology and the appropriate application of sets, functions and relations.

B. Constructs truth tables (for negation, conjunction, disjunction, implication, biconditional, bit operators) and uses them to demonstrate propositional relations.

C. Converts spoken language statements to appropriate statements in propositional logic.

D. Demonstrates proficiency in the use of Boolean algebra, including De Morgan's laws, to identify propositional equivalences.

E. Uses formal logic proofs and logical reasoning to solve problems and evaluate algorithmic complexity.

F. Computes permutations and combinations of a set and interprets the meaning in context.

G. Knows how to exhibit, describe and justify mathematical ideas and arguments through the use of precise mathematical language in written or oral communication.

Competency 011: *The computer science teacher knows digital forensics topics.*

The beginning teacher:

A. Understands that digital forensics involves the recovery and investigation of material found in digital devices, often in relation to computer crime.

B. Demonstrates knowledge of legal, illegal, ethical and unethical information-gathering methods; possible gray areas; and ways in which developing laws and guidelines affect digital forensics practices.

C. Understands that digital forensics involves the application of tools in a variety of investigations related to malicious attacks (e.g., worm infections, malware, phishing incidents, viruses, Trojans, rootkits, email threats).

D. Identifies and describes businesses and government agencies that use digital forensics.

NOTE: After clicking on a link, right click and select "Previous View" to go back to original text.

E. Demonstrates knowledge of how digital forensics fits in the workplace and how to establish guidelines, procedures and recommendations for the use of digital forensics tools.

F. Knows how to describe the function and use of digital forensics toolkits in the analysis of network traffic data and data files from various storage media.

Competency 012: *The computer science teacher knows robotics topics.*

The beginning teacher:

A. Demonstrates knowledge of technology concepts, systems and operations as they apply to robotics.

B. Knows how to utilize the design process to prototype, construct, evaluate, refine and document the development of a robot.

C. Demonstrates the use of computers to manipulate a robot.

D. Develops algorithms to move and provide interaction with a robot, including applying instructions, collecting sensor data and performing simple tasks (e.g., following lines, moving objects, avoiding obstacles).

E. Knows how to explore the effects robots have on culture and society.

F. Knows how to use software applications to simulate the behavior of robots of varying complexity, present design concepts and test solution strategies.

Competency 013: *The computer science teacher knows game and mobile application development topics.*

The beginning teacher:

A. Demonstrates an understanding of the software-development process specifically applied to mobile and desktop game applications.

B. Demonstrates knowledge of the basic game design process and elements.

C. Knows the fundamentals of game art, including the look and feel, graphics coordinate system, basics of color theory and image rendering.

D. Demonstrates an understanding of the user experience and knows how to create effective user interfaces, game rules and instructions.

E. Knows how to use board games to research and collect game-play data.

F. Demonstrates an understanding of game programming essentials, including event-driven programming and collision detection.

G. Knows how to use a simulation tool to imitate a mobile device’s functionality.

Approaches to Answering Selected-Response Questions

The purpose of this section is to describe selected-response question formats that you will typically see on the Computer Science 8–12 test and to suggest possible ways to approach thinking about and answering them. These approaches are intended to supplement and complement familiar test-taking strategies with which you may already be comfortable and that work for you. Fundamentally, the most important component in assuring your success on the test is knowing the content described in the test framework. This content has been carefully selected to align with the knowledge required to begin a career as a Computer Science 8–12 teacher.

The selected-response questions on this test are designed to assess your knowledge of the content described in the test framework. In most cases, you are expected to demonstrate more than just your ability to recall factual information. You may be asked to think critically about the information, to analyze it, consider it carefully and compare it with other knowledge you have or make a judgment about it.

Leave no questions unanswered. Questions for which you mark no answer are counted as incorrect. Your score will be determined by the number of questions you answer correctly.

The Computer Science 8–12 test is designed to include a total of 100 selected-response questions, out of which 80 are scored. Your final scaled score will be based only on scored questions. The questions that are not scored are being pilot tested to collect information about how these questions will perform under actual testing conditions. These pilot questions are not identified on the test.

How to Approach Unfamiliar Question Formats

Some questions include introductory information such as a code segment, table, graph or reading passage (often called a stimulus) that provides the information the question asks for. New formats for presenting information are developed from time to time. Tests may include audio and video stimulus materials such as a movie clip or some kind of animation, instead of a map or reading passage. Other tests may allow you to zoom in on the details in a graphic or picture.

Tests may also include interactive types of questions. These questions take advantage of technology to assess knowledge and skills that go beyond what can be assessed using standard single-selection selected-response questions. If you see a format you are not familiar with, **read the directions carefully**. The directions always give clear instructions on how you are expected to respond.

NOTE: After clicking on a link, right click and select "Previous View" to go back to original text.

For most questions, you will respond by clicking an oval to choose a single answer choice from a list of options. Other questions may ask you to respond by:

- **Selecting one or more correct answers.** In some questions you will be asked to choose a given number of correct answers, or all the options that answer the question correctly.
- **Typing in an entry box.** When the answer is a number, you might be asked to enter a numeric answer or, if the test has an on-screen calculator, you might need to transfer the calculated result from the calculator into the entry box. Some questions may have more than one place to enter a response.
- **Clicking check boxes.** You may be asked to click check boxes instead of an oval when more than one choice within a set of answers can be selected.
- **Clicking parts of a graphic.** In some questions, you will choose your answer by clicking on location(s) on a graphic such as a map or chart, as opposed to choosing from a list.
- **Clicking on sentences.** In questions with reading passages, you may be asked to choose your answer by clicking on a sentence or sentences within the reading passage.
- **Dragging and dropping answer choices into "targets" on the screen.** You may be asked to choose an answer from a list and drag it into the appropriate location in a table, paragraph of text or graphic.
- **Selecting options from a drop-down menu.** This type of question will ask you to select the appropriate answer or answers by selecting options from a drop-down menu (e.g., to complete a sentence).

Remember that with every question, you will get clear instructions on how to respond.

Question Format

You may see the following types of questions on the test:

— Single Questions
— Questions with Stimulus Materials (Code Segments)

On the following pages, you will find descriptions of these commonly used question formats, along with suggested approaches for responding to each type.

Single Questions

The single-question format presents a direct question or an incomplete statement. It can also include a code segment, reading passage, graphic, table or a combination of these. Four or more answer options appear below the question.

The following question is an example of the single-question format. It tests knowledge of Computer Science 8–12 Competency 003: *The computer science teacher knows how to plan, organize, deliver and evaluate instruction that effectively utilizes current technology for teaching the Technology Applications Texas Essential Knowledge and Skills (TEKS) to all students*.

Example 1

1. The most effective method of helping students improve their comprehension of the concepts and constructs of a specific programming language is to

   A. have the students memorize the syntax of the language.

   B. require the students to read several books about the language.

   C. give the students quizzes on a variety of topics related to the language.

   D. ask the students to write and test several programs using the language.

Suggested Approach

Read the question carefully and critically. Think about what it is asking and the situation it is describing. Eliminate any obviously wrong answers, select the correct answer choice and mark your answer.

In this situation, think about the most effective method of helping students improve their comprehension of the concepts and constructs of a specific programming language. Now look at the answer options and consider which describes the most effective method.

Option A indicates that having students memorize the syntax of a programming language is the most effective way of helping them improve their comprehension of its concepts and constructs. Knowledge of the syntax of a programming language is important. However, memorizing syntax is not effective in improving comprehension of the language. Therefore, option A is not an appropriate response.

Option B indicates that having students read several books on a programming language is the most effective way of helping them improve their comprehension of its concepts and constructs. Reading books on the language is worthwhile. However, doing so is not an extremely effective way of improving comprehension of the concepts and constructs of the language. Therefore, option B is not an appropriate response.

Option C indicates that giving students quizzes on several topics related to a programming language is the most effective way of helping them improve their comprehension of its concepts and constructs. Testing students on their knowledge of the language is worthwhile. However, doing so in the form of quizzes is not effective in helping them improve their comprehension of the concepts and constructs of the language. Therefore, option C is not an appropriate response.

Option D indicates that asking students to write and test several programs using the language is the most effective way of helping them improve their comprehension of its concepts and constructs. When writing and testing programs, students use the programming concepts they have learned, improving their understanding of those concepts. Therefore, option D is an appropriate response.

Of the alternatives offered, having students write and test several programs using a particular programming language is the most effective way of helping them improve their comprehension of the language's concepts and constructs. Therefore, **the correct response is option D**.

Example 2

The following question tests knowledge of Computer Science 8–12 Competency 004: *The computer science teacher knows problem-solving strategies and different procedures for program design*.

2. Which of the following distinguishes an object-oriented programming design strategy from other design strategies?

   A. Modules are tested individually and then together.

   B. A requirements specification is created as a first step.

   C. Functions and data are treated as integrated components.

   D. An incremental approach is used in the system development.

Suggested Approach

Read the question carefully and critically. Think about what it is asking and the situation it is describing. Eliminate any obviously wrong answers, select the correct answer choice and mark your answer.

In this situation, think about the design characteristics of an object-oriented programming design strategy that are distinct from those of a procedural programming design strategy. Now look at the answer options and consider which best describes one such distinction.

Option A suggests that a feature that distinguishes an object-oriented programming design strategy from other design strategies is that modules are tested individually and then together. This is a feature of object-oriented programming design strategies. However, it is also a feature of other programming design strategies, including procedural programming designs. Therefore, option A is not a feature that distinguishes an object-oriented programming design strategy from other design strategies.

Option B suggests that a feature that distinguishes an object-oriented programming design strategy from other design strategies is that a requirements specification is created as a first step in the design. This is a feature of object-oriented programming design strategies. However, it is also a feature of other programming design strategies, including procedural programming designs. Therefore, option B is not a feature that distinguishes an object-oriented programming design strategy from other design strategies.

Option C suggests that a feature that distinguishes an object-oriented programming design strategy from other design strategies is that functions and data are treated as integrated components. This is a feature of object-oriented programming design strategies that is not present in other programming design strategies, including procedural programming designs. Therefore, option C is a feature that distinguishes an object-oriented programming design strategy from other design strategies.

Option D suggests that a feature that distinguishes an object-oriented programming design strategy from other design strategies is that an incremental approach is used in the system development. This is a feature of object-oriented programming design strategies. However, it is also a feature of other programming design strategies, including procedural programming designs. Therefore, option D is not a feature that distinguishes an object-oriented programming design strategy from other design strategies.

Of the alternatives offered, only treating functions and data as integrated components could be considered a distinguishing feature of an object-oriented programming design strategy. Therefore, **the correct response is option C**.

Questions With Stimulus Material

Some questions on this test are preceded by stimulus material that relates to the question. Some types of stimulus material included on the test are reading passages, code segments, graphics, tables or a combination of these. In such cases, you will generally be given information followed by an event to analyze, a problem to solve or a decision to make.

You can use several different approaches to respond to questions with stimulus material. Some commonly used strategies are listed below.

Strategy 1 Skim the stimulus material to understand its purpose, its arrangement and/or its content. Then read the questions and refer again to the stimulus material to obtain the specific information you need to answer the questions.

Strategy 2 Read the questions *before* considering the stimulus material. The theory behind this strategy is that the content of the questions will help you identify the purpose of the stimulus material and locate the information you need to answer the questions.

NOTE: After clicking on a link, right click and select “Previous View” to go back to original text.

Strategy 3 Use a combination of both strategies. Apply the "read the stimulus first" strategy with shorter, more familiar stimuli and the "read the questions first" strategy with longer, more complex or less familiar stimuli. You can experiment with the sample questions in this manual and then use the strategy with which you are most comfortable when you take the actual test.

Whether you read the stimulus before or after you read the questions, you should read it carefully and critically. You may want to note its important points to help you answer the questions.

As you consider questions set in educational contexts, try to enter into the identified teacher's frame of mind and use that teacher's point of view to answer the questions that accompany the stimulus. Be sure to consider the questions only in terms of the information provided in the stimulus — not in terms of your own experiences or individuals you may have known.

Some stimulus material contains code segments written in pseudocode. The notation used in the pseudocode is described below.

PSEUDOCODE NOTATION

Explanation	**Notation**
Assignment operator	←
Arithmetic operators	`+ - / * ^ %`
Relational operators	`==` `<` `>` ≤ ≥ ≠
Logical operators	**and** **or** **not**
String concatenation operator	`+`
Boolean values	**true** **false**
Null	**null**

Comments	`// this is a single-line comment`
	
	`/* this is a C-style comment */`

Pre- and post-increment	`++i` `i++`
Print A comment is used to describe if a line feed or blank is printed between arguments, where necessary.	**print** arg **print** (arg) **print** (arg1, arg2, ...)

NOTE: After clicking on a link, right click and select "Previous View" to go back to original text.

Data types	**boolean** **char** **double** **float** **int** **int[]** **int[][]** **int array** **short** **String**
Array initialization and reference	**int[]** a ← {1, 2, 3} **int array** b ← {1, 2, 3} **int** c[0..2] ← {1, 2, 3} **int[][]** d a[0]

Conditional statements: Indentation and **end if** are significant; no braces used.
 	
Example:
if (x > 10)
    print ("big number")
else
    print ("small number")
end if
	if (condition)
	    block of statements
	end if
	
	if (condition)
	    block of statements
	else
	    another block of statements
	end if

Iterative statements: Indentation and **end** statements are significant.	**for** (initialization; condition; increment)
	{
	    block of statements
	}
	
	for (initialization; condition; increment)
	    block of statements
	end for
	
	while (condition)
	    block of statements
	end while
	
	do {
	    block of statements
	} **while** (condition)
	
	repeat
	    block of statements
	until (condition)

NOTE: After clicking on a link, right click and select "Previous View" to go back to original text.

Procedures: Indentation and **end** statements are significant; no braces are used. The return type is indicated in the procedure header based on the **return** statement in the procedure. When the return type is not indicated, the return type is **void**.	**int** procedureName (arg1, arg2, ...) block of statements **return** value **end** procedureName **procedure** procedureName (arg1, arg2, ...) block of statements **end procedure** procedureName
Parameter passing Example: **int** p (**pass-by-reference** x, **pass-by-value** y)	**pass-by-reference** **pass-by-value**
Classes	**class** className variable declarations procedures **end class** className
Object-oriented keywords	**extends** **new** **public** **private**

LOGIC NOTATION

Explanation	Notation
Conjunction, and	$\wedge$
Disjunction, or	$\vee$
Negation, not	$\neg$
Implication, implies	$\Rightarrow$
Biconditional, is equivalent	$\Leftrightarrow$
Universal quantifier, for all	$\forall$
Existential quantifier, there exists	$\exists$

NOTE: After clicking on a link, right click and select "Previous View" to go back to original text.

Example

The following question contains a code segment as stimulus material. The question measures the knowledge of Computer Science 8–12 Competency 008: *The computer science teacher correctly and efficiently uses statements and control structures in the development of code*.

Consider the following pseudocode segment.

```
int k ← 18
for ( int j ← 1; j ≤ 5; j ← j + 1 )
    if ( k > 10 )
        /* missing if body */
    end if
end for
```

If the value of `k` at the end of the code segment is `10`, which of the following expressions should replace `/* missing if body */`?

A. `k ← k - 1`

B. `k ← k - 3`

C. `k ← k - 4`

D. `k ← k - 6`

Suggested Approach

Examine carefully the pseudocode segment presented in the stimulus; then read the question. Now substitute each response choice and hand trace each resulting code segment to determine which one yields a value of `10` for the variable `k`. The initial value of `k` is `18`. The index variable `j` in the `for` statement increases from `1` to `5` by `1`.

Option A, which indicates that `k ← k - 1` should be the missing code, would result in a value of `13` for `k`. Thus, option A is an incorrect response.

Option B, which indicates that `k ← k - 3` should be the missing code, would result in a value of `9` for `k`. Thus, option B is an incorrect response.

Option C, which indicates that `k ← k - 4` should be the missing code, would result in a value of `10` for `k`. Thus, option C is a correct response.

Option D, which indicates that `k ← k - 6` should be the missing code, would result in a value of `6` for `k`. Thus, option D is an incorrect response.

NOTE: After clicking on a link, right click and select "Previous View" to go back to original text.

Of the alternatives offered, only `k ← k - 4` results in a value of `10` for `k`. Therefore, **the correct response is option C**.

CODE SEGMENTS

The following are some examples of pseudocode stimulus material.

Example 1

Class declaration and object instantiation

```
class StudentInfo
    int studentID
    String name
end class StudentInfo

StudentInfo x ← new StudentInfo ()
x.studentID ← 1234
    // the value 1234 is assigned to x.studentID
x.name ← "John"
print ( x.studentID )
print ( x.name )
```

Example 2

The following procedure uses different parameter-passing mechanisms for the two parameters.

```
void f ( pass-by-reference int x, pass-by-value int y )
    x ← y + 1
    y ← x + 2
end f
```

NOTE: After clicking on a link, right click and select "Previous View" to go back to original text.

Example 3

Insertion sort

```
// precondition 1: A is an array of integers.
// precondition 2: The length of array A is n.
// precondition 3: The index of array A starts at 0.
int[] insertionSort ( pass-by-reference int[] A, int n )
    for ( int j ← 1; j ≤ n - 1; j ← j + 1 )
        int temp ← A[j]
        int k ← j - 1
        while ( ( k ≥ 0 ) and ( A[k] > temp ) )
            A[k + 1] ← A[k]
            k ← k - 1
        end while
        A[k + 1] ← temp
    end for
    return A // returns the sorted array
end insertionSort
```

NOTE: After clicking on a link, right click and select "Previous View" to go back to original text.

Selected-Response Practice Questions

This section presents some sample test questions for you to review as part of your preparation for the test. To demonstrate how each competency may be assessed, each sample question is accompanied by the competency that it measures. While studying, you may wish to read the competency before and after you consider each sample question. Please note that the competency statements do not appear on the actual test.

For each sample test question, there is at least one correct answer and a rationale for each answer option. Please note that the sample questions are not necessarily presented in competency order.

The sample questions are included to illustrate the formats and types of questions you will see on the test; however, your performance on the sample questions should not be viewed as a predictor of your performance on the actual test.

NOTE: After clicking on a link, right click and select "Previous View" to go back to original text.

COMPETENCY 001

1. Which of the following is the principal advantage of saving a word processing document in rich-text format?

   A. The document can be viewed in any Web browser.
   B. A formatted document can be transferred between different applications.
   C. The document can take up less space in memory.
   D. A formatted document can be scanned for viruses when sent as an email attachment.

Answer and Rationale

COMPETENCY 001

2. Which of the following would most likely be considered unacceptable use of information by a teacher?

   A. Using the school district's database to determine gender distribution in local schools
   B. Using the Internet history on a classroom computer to audit student Internet use
   C. Using students' personal data to create a mailing list for a local charity
   D. Using classroom records to determine recipients of academic awards

Answer and Rationale

NOTE: After clicking on a link, right click and select "Previous View" to go back to original text.

COMPETENCY 001

3. Consider the uniform resource locator (URL) `https://example.net/index.html`. Which of the following are contained in the URL?

   Select all that apply.

   A. Browser name
   B. Email address
   C. File name
   D. Host name
   E. MAC address
   F. Protocol

Answer and Rationale

COMPETENCY 002

4. Students in a Texas classroom have been communicating with a class in New York by videoconference. The two classes find that the images they receive from each other occasionally freeze for up to 30 seconds before the video continues. This type of problem can most often be solved by

   A. increasing bandwidth.
   B. upgrading cameras.
   C. increasing video resolution.
   D. upgrading monitors.

Answer and Rationale

COMPETENCY 002

5. Which of the following is the most appropriate format for graphics that are to be embedded within an Internet document?

   A. BMP
   B. TIFF
   C. PNG
   D. HTML

Answer and Rationale

NOTE: After clicking on a link, right click and select "Previous View" to go back to original text.

COMPETENCY 002

6. Suppose that the class grade for a six-week period is based on 3 tests (T1, T2, T3), each of which counts for 15%, 4 quizzes (Q1, Q2, Q3, Q4), each of which counts for 10%, and a homework notebook (HW), which counts for 15%. The grades are recorded in a spreadsheet similar to the one below.

	A	B	C	D	E	F	G	H	I	J
1	Name	T1	T2	T3	Q1	Q2	Q3	Q4	HW	AVG
2	Jane	87	92	80	76	79	87	74	90	
3	Joe	91	85	77	78	88	96	90	92	
4	Bill	65	72	70	80	81	74	77	80	
5	Brenda	96	88	91	76	91	100	74	98	

Which of the following formulas would be a correct calculation of the six-week weighted average for Jane?

A. =(B2+C2+D2+E2+F2+G2+H2+I2)/8
B. =(B2+C2+D2+I2)*0.15+(E2+F2+G2+H2)*0.1
C. =(B2+C2+D2+I2)*15+(E2+F2+G2+H2)*10
D. =(B2+C2+D2+I2)/15+(E2+F2+G2+H2)/10

Answer and Rationale

COMPETENCY 003

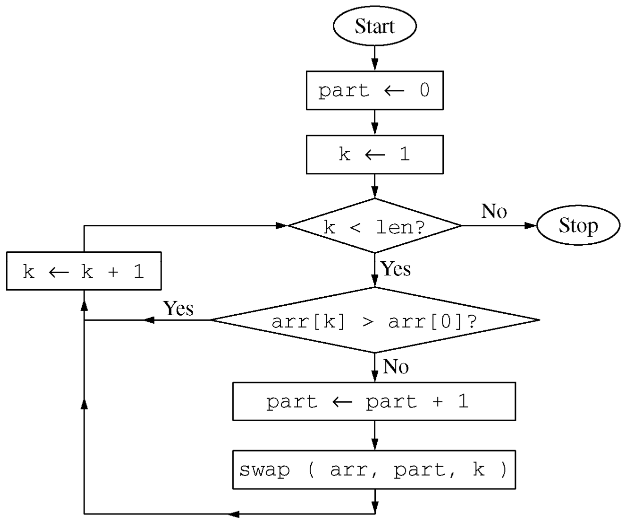
7. A teacher has assigned students several topics to discuss outside of class using an electronic form of communication. The teacher wants the students' messages to be organized by topic and wants to have all historical messages available to students. To facilitate this type of communication most effectively, the teacher should have students

A. participate in a threaded discussion group.
B. send email messages with attached document files.
C. update pages on the class's Web site.
D. engage in dialogue in a real-time chat room.

Answer and Rationale

NOTE: After clicking on a link, right click and select "Previous View" to go back to original text.

8. Consider the following flowchart diagram, where `arr[0..len-1]` is an integer array of length `len`. Assume that the elements `arr[0]`, `arr[1]`, ..., `arr[len-1]` have already been initialized.



Which of the following pseudocode segments implements the algorithm in the flowchart?

A.
```
int part ← 0
int k ← 1
while ( k < len )
    if ( arr[k] ≤ arr[0] )
        part ← part + 1
        swap ( arr, part, k )
    end if
    k ← k + 1
end while
```

B.
```
int part ← 0
int k ← 1
while ( k < len )
    k ← k + 1
    if ( arr[k] ≤ arr[0] )
        part ← part + 1
        swap ( arr, part, k )
    end if
end while
```

NOTE: After clicking on a link, right click and select “Previous View” to go back to original text.

C.
```
int part ← 0
int k ← 1
while ( k < len )
    if ( arr[k] > arr[0] )
        part ← part + 1
        swap ( arr, part, k )
    end if
    k ← k + 1
end while
```

D.
```
int part ← 0
int k ← 1
while ( k < len )
    k ← k + 1
    if ( arr[k] > arr[0] )
        part ← part + 1
        swap ( arr, part, k )
    end if
end while
```

Answer and Rationale

COMPETENCY 004

9. Which of the following best describes the purpose of generating a flowchart as part of the design of a computer program?

A. To test and maintain the efficiency of the overall program
B. To present the steps needed to solve the programming problem
C. To ensure that all methods are appropriately linked
D. To determine the necessary number of global and local variables

Answer and Rationale

NOTE: After clicking on a link, right click and select "Previous View" to go back to original text.

COMPETENCY 005

10. A software system is to be developed for which the requirements are well understood and the risk of failure is minimal. To meet these requirements, which of the following software development models would be most appropriate to use?

    A. Chaos
    B. Incremental
    C. Spiral
    D. Waterfall

Answer and Rationale

COMPETENCY 005

11. The most appropriate way to use a library of program code is to access the

    A. methods or functions by way of the interface.
    B. implementation details of the methods or functions.
    C. methods or functions by way of the source code.
    D. documentation of the methods or functions.

Answer and Rationale

NOTE: After clicking on a link, right click and select "Previous View" to go back to original text.

COMPETENCY 005

12. Consider the following pseudocode segment with integer variables, where the precondition at the beginning of the segment is missing.

```
// missing precondition
x ← x + 1
y ← y + x
// postcondition:  y == 2 * x
```

Which of the following would be a valid precondition for the code segment above?

A. `y == x - 1`
B. `y == x`
C. `y == x + 1`
D. `y == x + 2`

Answer and Rationale

COMPETENCY 006

13. Which of the following techniques is used by most programming languages to intercept events that disrupt the normal flow of a program's execution?

A. Code security
B. Flow control
C. Exception handling
D. Error detection

Answer and Rationale

NOTE: After clicking on a link, right click and select "Previous View" to go back to original text.

COMPETENCY 006

14. If execution speed and direct communication with devices such as controllers and processors are essential to the success of a project, which of the following programming languages would be most appropriate to use?

    A. C
    B. Java
    C. PHP
    D. Visual Basic

Answer and Rationale

COMPETENCY 007

15. Which of the following is most efficient for manipulating a list that contains integers and is of predefined size?

    A. A stack
    B. A linked list
    C. An array
    D. A sequential file

Answer and Rationale

COMPETENCY 007

16. A programmer is developing a program to read strings from a file and store the strings in a data structure. The strings are unordered in the file but must be accessible in alphabetical order in the data structure. The program must also be able to add and remove strings from the data structure.

    Which of the following data structures is the best choice for the program so that the requirements for creating the data structure, adding and removing elements and accessing elements in sorted order are met as efficiently as possible?

    A. A binary search tree
    B. A linked list
    C. A queue
    D. A stack

Answer and Rationale

NOTE: After clicking on a link, right click and select “Previous View” to go back to original text.

COMPETENCY 007

17. Consider the following pseudocode procedure `calc`, where the first and second parameters are passed by value and the third and fourth parameters are passed by reference. That is, actual parameters passed to formal parameters `w` and `x` are passed by value, while those passed to formal parameters `y` and `z` are passed by reference.

```
procedure calc ( pass-by-value int w,
                 pass-by-value int x,
                 pass-by-reference int y,
                 pass-by-reference int z )
     w ← w + 1
     x ← x * 2
     y ← y + 3
     z ← z * 4
end procedure
```

What are the values of `a` and `b` at the end of the code fragment below?

```
int a ← 5
int b ← 6
calc ( a, a, b, b )
```

A. `a = 5` and `b = 24`

B. `a = 5` and `b = 36`

C. `a = 10` and `b = 6`

D. `a = 12` and `b = 6`

Answer and Rationale

NOTE: After clicking on a link, right click and select "Previous View" to go back to original text.

COMPETENCY 007

18. Consider a class `Stack` defined with methods `push ( x )`, `pop ()` and `peek ()` that implement a stack data structure. (Note that **`void`** `push (` **`int`** `x )` pushes the integer `x` onto the top of the stack; **`int`** `pop ()` removes the integer at the top of the stack and returns that integer; **`int`** `peek ()` returns the integer at the top of the stack without removing it from the stack.)

    Consider the following pseudocode fragment, where `S` is a `Stack` instance that will hold integers.

```
Stack S ← new Stack ()
S.push ( 4 )
S.push ( 3 )
S.push ( S.peek () + S.peek () )
S.push ( S.pop () * S.pop () )
print ( S.peek () )
```

    What is printed by the last line of code?

    A. 18
    B. 21
    C. 28
    D. 32

Answer and Rationale

NOTE: After clicking on a link, right click and select "Previous View" to go back to original text.

COMPETENCY 008

19. Consider the following pseudocode functions, where each **print** statement prints on a separate line of output and then executes a line feed.

```
void f1 ( int n )
    int k ← 0
    do {
        k ← k + 1
        print k
    } while ( k < n )
end f1


void f2 ( int n )
    int k ← 0
    while ( k < n )
        k ← k + 1
        print k
    end while
end f2
```

Which of the following describes all the values of the input `n` for which functions `f1` and `f2` print the same sequence of numbers?

A. $n > 0$

B. $n \geq 0$

C. $n < 0$

D. $n \leq 0$

Answer and Rationale

NOTE: After clicking on a link, right click and select "Previous View" to go back to original text.

COMPETENCY 008

20. Consider the following pseudocode fragment, where `x` is an integer variable initialized to a nonnegative integer value.

```
// x is a nonnegative integer
int sum
x ← x / 2    // integer division; truncates fractions
for ( sum ← 1; x > 0; x ← x / 2 )
    sum ← sum + 1
end for
```

Which of the following will calculate the same value of `sum` as the fragment above?

A.
```
int sum ← 0
x ← x / 2
while ( x ≥ 0 )
    sum ← sum + 1
    x ← x / 2
end while
```

B.
```
int sum ← 1
x ← x / 2
while ( x ≥ 0 )
    sum ← sum + 1
    x ← x / 2
end while
```

C.
```
int sum ← 0
do {
    sum ← sum + 1
    x ← x / 2
} while ( x > 0 )
```

D.
```
int sum ← 1
do {
    sum ← sum + 1
    x ← x / 2
} while ( x > 0 )
```

Answer and Rationale

NOTE: After clicking on a link, right click and select "Previous View" to go back to original text.

COMPETENCY 008

21. Consider the following pseudocode fragment with integer variables.

```
total ← 0
x ← 1

while ( x < ( 2 * n ) )
    if ( ( x % 2 ) == 1 ) // if x is odd
        total ← total + x
    end if
    x ← x + 1
end while

print ( total )
```

Assume that n has been initialized with a positive integer value. What value is printed when the code fragment is executed?

A. 0
B. n
C. 2n
D. n^2

Answer and Rationale

COMPETENCY 009

22. Which of the following represents the average-case performance of a quicksort algorithm?

A. $O(n)$
B. $O(\log_2 n)$
C. $O\left(n^2\right)$
D. $O(n \log_2 n)$

Answer and Rationale

NOTE: After clicking on a link, right click and select "Previous View" to go back to original text.

COMPETENCY 009

23. Consider the following pseudocode function, where each **`print`** statement prints on a separate line of output and then executes a line feed.

```
void h ( int n )
    if ( n ≥ 4 )
        h ( n / 2 )
    end if
    print n
end h
```

What is printed when the call `h ( 16 )` is executed?

A. 2

B. 16

C. 16
8
4
2

D. 2
4
8
16

Answer and Rationale

COMPETENCY 009

24. A specific sorting algorithm begins by finding the largest element and swapping that element with the last element. Which of the following sorting algorithms fits this description?

A. Quicksort

B. Insertion sort

C. Heapsort

D. Selection sort

Answer and Rationale

NOTE: After clicking on a link, right click and select "Previous View" to go back to original text.

25. Consider the following pseudocode binary search function, which returns the largest array index `k` such that `a[k] ≤ x`.

```
// precondition 1:  integer array a is sorted in
//                  ascending order
// precondition 2:  0 ≤ first < last < length of array a
// precondition 3:  a[first] ≤ x < a[last]
int f(int array a, int x, int first, int last)
    while ( first + 1 ≠ last )
        int mid ← ( first + last ) / 2    // integer division
        if ( x < a[mid] )
            last ← mid
        else
            first ← mid
        end if
    end while
    return first
end f
```

Consider the following (incomplete) equivalent recursive implementation of function `f`.

```
int f(int array a, int x, int first, int last)
    if ( first + 1 == last )
        return first
    end if
    int mid ← ( first + last ) / 2
    // missing code block
end f
```

Which of the following could replace the missing code block so that the recursive function will work as intended?

A.
```
if ( x ≥ a[mid] )
    return f ( a, x, first, mid )
end if
return f ( a, x, mid, last )
```

B.
```
if ( x ≥ a[mid] )
    return f ( a, x, mid, first )
end if
return f ( a, x, last, mid )
```

NOTE: After clicking on a link, right click and select "Previous View" to go back to original text.

C.
```
if ( x ≥ a[mid] )
     return f ( a, x, mid, last )
end if
return f ( a, x, first, mid )
```

D.
```
if ( x ≥ a[mid] )
     return f ( a, x, last, mid )
end if
return f ( a, x, mid, first )
```

Answer and Rationale

COMPETENCY 009

26. Consider the following pseudocode function.

```
// precondition:  n and k are nonnegative integers
int f ( int n, int k )
    if ( k * n == 0 )
        return 1
    else
        return f ( n - 1, k - 1 ) + f ( n - 1, k )
    end if
end f
```

What value is returned by the call `f ( 4, 2 )` ?

A. 4

B. 5

C. 7

D. 11

Answer and Rationale

NOTE: After clicking on a link, right click and select "Previous View" to go back to original text.

COMPETENCY 009

27. Consider the following pseudocode segment with integer variables that implements a selection sort. Assume that `A` is an integer array of length `n` with indexing that starts at `0`.

```
for ( int j ← 0; j < n - 1; j ← j + 1 )
    int x ← j
    for ( int i ← j + 1; i < n; i ← i + 1 )
        // missing code block
    end for
    if ( x ≠ j )
        swap ( A[x], A[j] )     // swap the two array entries
    end if
end for
```

Which of the following could replace the missing code block so that the code segment will work as intended?

A.
```
if ( A[x] > A[i] )
    x ← i
end if
```

B.
```
if ( A[x] > A[i] )
    A[x] ← A[i]
end if
```

C.
```
if ( x > i )
    x ← i
end if
```

D.
```
if ( x > i )
    A[x] ← A[i]
end if
```

Answer and Rationale

NOTE: After clicking on a link, right click and select “Previous View” to go back to original text.

COMPETENCY 010

28. Which of the following truth tables correctly represents the Boolean expression $(p \wedge q) \Leftrightarrow (p \vee q)$?

A.

p	q	$(p \wedge q) \Leftrightarrow (p \vee q)$
1	1	0
1	0	1
0	1	1
0	0	0

B.

p	q	$(p \wedge q) \Leftrightarrow (p \vee q)$
1	1	1
1	0	0
0	1	0
0	0	1

C.

p	q	$(p \wedge q) \Leftrightarrow (p \vee q)$
1	1	1
1	0	1
0	1	1
0	0	0

D.

p	q	$(p \wedge q) \Leftrightarrow (p \vee q)$
1	1	1
1	0	0
0	1	0
0	0	0

Answer and Rationale

NOTE: After clicking on a link, right click and select "Previous View" to go back to original text.

COMPETENCY 010

29. Consider propositions *p* and *q*, defined as follows.

    p: I go for a run.
    q: The sky is dark.

    Which of the following is equivalent to the compound proposition "I don't go for a run when the sky is dark"?

    A. $p \Rightarrow q$

    B. $\neg p \Rightarrow q$

    C. $q \Rightarrow \neg p$

    D. $\neg q \Rightarrow p$

Answer and Rationale

COMPETENCY 011

30. Which of the following describes how data remanence is relevant to digital forensics?

    A. It allows the recovery of digital data even though file deletion has occurred.
    B. It determines whether data are preserved or lost when a computer is turned off.
    C. It ensures that collected evidence is admissible as evidence in a court of law.
    D. It establishes who has authorization to monitor and collect network traffic data.

Answer and Rationale

NOTE: After clicking on a link, right click and select "Previous View" to go back to original text.

COMPETENCY 012

31. A robot's programming system uses the command `move[motor] ← value`, where `motor` identifies a particular motor and `value` is an integer amount of speed, with a positive value indicating forward movement, a negative value indicating backward movement and 0 indicating a stop. For example, the command `move[left] ← 99` will cause the left motor to move forward at a speed of 99.

A student is programming a two-wheel-drive robot to travel through a maze and is having trouble with the corners. The robot swings wide and goes out of bounds. A segment of the code being used for a right turn is similar to the code below, where `left` is the left wheel motor (from the robot's perspective), `right` is the right wheel motor and `slow` is a positive integer representing an appropriate turning speed.

```
move[left] ← slow
move[right] ← 0
```

The teacher suggests that the student consider modifying the robot's turning code to execute a point (in-place) turn rather than a swing turn. Which of the following code segments could the student use to implement the teacher's suggestion?

A.
```
move[left] ← slow
move[right] ← -slow
```

B.
```
move[left] ← 0
move[right] ← slow
```

C.
```
move[left] ← slow
move[right] ← 2 * slow
```

D.
```
move[left] ← -slow
move[right] ← slow
```

Answer and Rationale

NOTE: After clicking on a link, right click and select "Previous View" to go back to original text.

32. Consider a game in a two-dimensional space and the goal of determining whether a collision has occurred between two circular objects (that is, to detect whether two circular objects overlap or touch). The centers of the circular objects are stored in variables `(x1, y1)` and `(x2, y2)` and the radii are stored in variables `r1` and `r2`. The distance between the two centers is given by the formula $\sqrt{(x1-x2)^2+(y1-y2)^2}$.

The following pseudocode segment is intended to implement a collision-detection algorithm.

```
collision ← false
// dist is the distance between centers.
dist ← sqrt ( ( x1 - x2 )^2 + ( y1 - y2 )^2 )
if ( <missing condition> )
    collision ← true
end if
```

Which of the following could replace `<missing condition>` so that the collision detection algorithm works as intended?

A. `dist ≥ r1 - r2`

B. `dist ≤ r1 + r2`

C. `( dist ≤ r1 ) or ( dist ≤ r2 )`

D. `( dist ≤ r1 ) and ( dist ≤ r2 )`

Answer and Rationale

NOTE: After clicking on a link, right click and select "Previous View" to go back to original text.

Answer Key and Rationales

Question Number	Competency Number	Correct Answer	Rationales
1	001	B	**Option B is correct** because most word processing applications can read and write rich-text format documents. **Options A is incorrect** because typical Web browsers do not support rich-text documents directly. **Option C is incorrect** because rich-text documents take up more space in memory than the corresponding documents in a plain text format or in the native format of the word processing application. **Option D is incorrect** because other formats can be scanned for viruses as well. Back to Question
2	001	C	**Option C is correct** because it fails to protect personally identifiable information. **Option A is incorrect** because gender (by itself, without any other additional information) is not considered personally identifiable information and because aggregate statistics are computed. **Option B is incorrect** because students should have no expectations of privacy when accessing Internet content using a classroom computer. The school can monitor the network usage in order to determine compliance with acceptable use guidelines. **Option D is incorrect** because classroom records are appropriate sources to use when selecting winners of academic awards. Back to Question

Question Number	Competency Number	Correct Answer	Rationales
3	001	C, D, F	**Options C, D and F are correct. Option C is correct** because the file name is indicated by `index.html`. **Option D is correct** because the hostname is indicated by `example.net`. **Option F is correct** because the protocol is indicated by `https`. **Options A, B and E are incorrect** because the URL does not contain information about browser name, email address or MAC address. Back to Question
4	002	A	**Option A is correct** because an increase in bandwidth will allow more data to be transferred and will help eliminate the freezing of the image. **Option B is incorrect** because updating cameras will not directly allow more data to be transferred. **Option C is incorrect** because increasing the video resolution will increase the amount of data to be transferred and could therefore cause the images to freeze for longer time periods. **Option D is incorrect** because upgrading monitors will not directly allow more data to be transferred. Back to Question
5	002	C	**Option C is correct** because PNG is a popular image format on the Internet because of its relatively small image size. Other Web-friendly image formats are GIF and JPEG. **Options A and B are incorrect** because BMP and TIFF images are typically very large. **Option D is incorrect** because HTML is not a format for graphics. Back to Question

Question Number	Competency Number	Correct Answer	Rationales
6	002	B	**Option B is correct** because it demonstrates the correct computation of a weighted mean, where each value is multiplied by the corresponding percent value and then the results are summed. **Option A is incorrect** because it calculates the mean, not the weighted average. **Option C is incorrect** because it is the correct result multiplied by 100. **Option D is incorrect** because it uses division rather than multiplication. Back to Question
7	003	A	**Option A is correct** because it meets both the requirement that the messages are organized by topic and the requirement that all old messages are available. **Option B is incorrect** because it does not meet either requirement. **Option C is incorrect** because updating a Web page to add a message is unnecessarily time-consuming and would likely lead to contention issues between students attempting to post messages simultaneously. **Option D is incorrect** because the messages will not be organized by topic. Back to Question

Question Number	Competency Number	Correct Answer	Rationales
8	004	A	**Option A is correct** because the code segment matches the steps in the flowchart. Two variables are initialized, followed by a **`while`** loop that is executed when `k < len`. Within the **`while`** loop, the condition on the **`if`** statement is the opposite of the corresponding condition in the flowchart, so `part` is updated and `swap` is called if the condition on the **`if`** statement is true. The variable `k` is incremented and the condition in the **`while`** loop is tested again. **Option B is incorrect** because the increment of `k` needs to occur after the **`if`** statement. **Option C is incorrect** because the increment of `part` and the call to the `swap` method need to occur when the comparison `arr[k] > arr[0]` is false. **Option D is incorrect** for both of the reasons stated in options B and C. Back to Question
9	004	B	**Option B is correct** because a flowchart is a graphic representation of a process, so it can be used to represent the steps in a computer program. **Option A is incorrect** because a flowchart is not a testing tool. **Options C and D are incorrect** because flowcharts cannot help to establish links between methods or analyze the variables used in a program. Back to Question

Question Number	Competency Number	Correct Answer	Rationales
10	005	D	**Option D is correct** because waterfall is typically used when requirements are well understood and the risk of failure is minimal. **Options A, B and C are incorrect** because those software development models are typically used in situations where the requirements are not well understood at the beginning of a project and change is anticipated. Back to Question
11	005	A	**Option A is correct** because the methods or functions in a library are accessed using an interface. **Options B, C and D are incorrect** because accessing the implementation details, source code or documentation provides information about how the methods or functions work but does not provide access to their functionality. Back to Question

Question Number	Competency Number	Correct Answer	Rationales

12	005	C	**Option C is correct** because the precondition `y == x + 1` is equivalent to the postcondition `y == 2 * x`. If `x_pre` and `y_pre` are the values of `x` and `y` before the code segment and if `x_post` and `y_post` are the values of `x` and `y` after the code segment, then we have the following relations.
			
			`x_post = x_pre + 1`
			`y_post = x_pre + y_pre + 1`
			
			When `x_post` and `y_post` replace `x` and `y` in the postcondition `y == 2 * x`, we get `x_pre + y_pre + 1 = 2 * (x_pre + 1)`, or equivalently `y_pre = x_pre + 1`. **Option A is incorrect** because the given precondition is equivalent to the postcondition `y == 2 * x - 2`. **Option B is incorrect** because the given precondition is equivalent to the postcondition `y == 2 * x - 1`. **Option D is incorrect** because the given precondition is equivalent to the postcondition `y == 2 * x + 1`.
			
			Back to Question

13	006	C	**Option C is correct** because exception handling is used to intercept events. **Options A, B and D are incorrect** because they refer to the presence of security holes in code, the flow of execution of programming statements and the detection (but not necessarily handling) of errors in a program. Back to Question

Question Number	Competency Number	Correct Answer	Rationales
14	006	A	**Option A is correct** because C is a relatively small, efficient programming language that can be used to communicate directly with various devices. **Options B, C and D are incorrect** because these languages are less appropriate to use when execution speed and direct communication with devices are essential. Back to Question
15	007	C	**Option C is correct** because an array is most appropriate for traversing and updating a list with the given conditions. **Options A and B are incorrect** because stacks and linked lists do not have predefined sizes; they are intended to grow and shrink. **Option D is incorrect** because a sequential file does not provide easy access to individual elements and modifying individual elements is difficult. Back to Question
16	007	A	**Option A is correct**. Of the data structures given the only one that maintains its elements in sorted order by default is the binary search tree. **Option B is incorrect**. A linked list can be used to maintain data in sorted order, but requires sequential search to find an item to remove, or to create the initial ordered list. **Options C and D are incorrect** because neither data structure facilitates the maintenance of a list in sorted order. Back to Question

Question Number	Competency Number	Correct Answer	Rationales
17	007	B	**Option B is correct** because at the end of the code fragment the values of `a` and `b` are `5` and `36`, respectively. Since the first two parameters are passed by value, the value of `a` after the `calc` call is the same as the value of `a` before the `calc` call. Since the last two parameters are passed by reference, the parameters `y` and `z` point to variable `b`. The value of `b` at the end of the code fragment is the value of `z` at the end of the procedure. Since the value of `b` is originally `6`, the value of `b` after `y ← y + 3` is `9`, and the value of `b` after `z ← z * 4` is `36`. **Option A is incorrect** because it fails to recognize that `z` has the value `9` when the statement `z = z * 4` is executed, using the original value of `6` instead. **Option D is incorrect** because it confuses the meaning of pass-by-reference and pass-by-value. **Option C is incorrect** because it results from the errors present in both options A and D. Back to Question

Question Number	Competency Number	Correct Answer	Rationales
18	007	A	**Option A is correct**. In the second and third lines of code, the values 4 and 3 are pushed onto the stack. In the fourth line of code, both peek operations return the value 3, so the value 6 is pushed onto the stack. In the fifth line of code, the two pop operations return 6 and 3, removing those values from the stack. Their product, 18, is then pushed onto the stack. The final line of code returns the value at the top of the stack, 18. **Options B, C and D are incorrect** because they correspond to incorrect arithmetic operations or misconceptions about stack methods. Back to Question
19	008	A	**Option A is correct** because when `n` is a positive integer the two functions print the same sequence of numbers. **Options B, C and D are incorrect** because when `n` is `0` or negative, the **`do-while`** loop in function `f1` is executed once and prints one number; the **`while`** loop in function `f2` is never entered and no numbers are printed. Back to Question
20	008	C	**Option C is correct** because the given **`do-while`** loop is equivalent to the given **`for`** loop. **Options A and B are incorrect** because the value of `x` will eventually become `0` and the **`while`** loop will loop forever. **Option D is incorrect** because the variable `sum` needs to be initialized to `0`. Back to Question

Question Number	Competency Number	Correct Answer	Rationales
21	008	D	**Option D is correct**. The code segment adds all the positive odd numbers less than 2n and prints the sum. Since $1+3+5+\ldots+(2n-1)=n^2$, the correct answer is D. **Options A, B and C are incorrect** because they do not represent the printed value. Back to Question
22	009	D	**Option D is correct** because on average the quicksort algorithm takes $O(n \log_2 n)$ comparisons to sort *n* items. **Options A, B and C are incorrect** because they are not equivalent to $O(n \log_2 n)$. Back to Question
23	009	D	**Option D is correct** because the call `h(16)` prints four lines of output containing the numbers `2, 4, 8` and `16`, respectively. It first calls `h(8)` and then will print `16` on a new line after the call `h(8)` completes. The call `h(8)` calls `h(4)` and then will print `8` on a new line after the call `h(4)` completes. The call `h(4)` calls `h(2)` and then will print `4` on a new line after the call `h(2)` completes. The call `h(2)` prints `2`, the first value printed. As each recursive call returns, the values `2, 4, 8` and `16` are printed. **Options A, B and C are incorrect** because they correspond to misconceptions about how recursive functions work. Back to Question

Question Number	Competency Number	Correct Answer	Rationales
24	009	D	**Option D is correct** because it describes how selection sort works. **Option A is incorrect** because quicksort uses a partition operation to divide the input array into two smaller sub-arrays and then sorts the sub-arrays recursively. **Option B is incorrect** because, in each iteration, insertion sort removes one element from the input, finds its correct location in the part already sorted and inserts it there. **Option C is incorrect** because heapsort uses a heap data structure. Back to Question
25	009	C	**Option C is correct** because, if `x ≥ a[mid]`, the value `x` could only be in `a[mid..last]`; otherwise, the value `x` could only be in `a[first..mid]`. **Options A, B and D are incorrect** because they correspond to misconceptions about how binary search works. Back to Question

Question Number	Competency Number	Correct Answer	Rationales

26	009	D	**Option D is correct** because the value returned by the call `f(4, 2)` is `11`.
			`f(4, 2)= f(3,1) + f(3,2)`
			`= f(2,0) + f(2,1) + f(2,1) + f(2,2)`
			`= 1 + 2 * f(2,1) + f(2,2)`
			`= 1 + 2 * (f(1,0) + f(1,1)) + (f(1,1) + f(1,2))`
			`= 3 + 3 * f(1,1)+ f(1,2)`
			`= 3 + 3 * (f(0,0) + f(0,1)) + (f(0,1) + f(0,2))`
			`= 11`
			Options A, B and C are incorrect because they correspond to misconceptions about how recursive functions work.
			
			Back to Question

27	009	A	**Option A is correct.** In each iteration of the outer **`for`** loop, the inner **`for`** loop identifies the smallest value in subarray `A[j+1..n-1]` and the **`if`** statement swaps it with `A[j]`. **Options B, C and D are incorrect** because they represent misconceptions about how selection sort works. Back to Question

<table>
<tr><th>Question Number</th><th>Competency Number</th><th>Correct Answer</th><th>Rationales</th></tr>
<tr><td>28</td><td>010</td><td>B</td><td>Option B is correct. The table below shows the complete truth table.
<table>
<tr><th>p</th><th>q</th><th>$p \wedge q$</th><th>$p \vee q$</th><th>$(p \wedge q) \Leftrightarrow (p \vee q)$</th></tr>
<tr><td>1</td><td>1</td><td>1</td><td>1</td><td>1</td></tr>
<tr><td>1</td><td>0</td><td>0</td><td>1</td><td>0</td></tr>
<tr><td>0</td><td>1</td><td>0</td><td>1</td><td>0</td></tr>
<tr><td>0</td><td>0</td><td>0</td><td>0</td><td>1</td></tr>
</table>
Option A is incorrect because it’s the negation of the correct values. Option C is incorrect because it results from using “or” instead of “if and only if.” Option D is incorrect because it results from using “and” instead of “if and only if.”

Back to Question</td></tr>
<tr><td>29</td><td>010</td><td>C</td><td>Option C is correct because the compounded statement is equivalent to “If the sky is dark, then I don’t go for a run,” which is represented by $q \Rightarrow \neg p$. Option A is incorrect because $p \Rightarrow q$ is equivalent to “If I go for a run, then the sky is dark.” Option B is incorrect because $\neg p \Rightarrow q$ is equivalent to “If I don’t go for a run, then the sky is dark.” Option D is incorrect because $\neg q \Rightarrow p$ is equivalent to “If the sky is not dark, then I go for a run.”

Back to Question</td></tr>
</table>

Question Number	Competency Number	Correct Answer	Rationales
30	011	A	**Option A is correct** because data remanence refers to the data that remain on a storage device after data are deleted. **Options B, C and D are incorrect** because they do not describe how data remanence is relevant to digital forensics. Back to Question
31	012	A	**Option A is correct** because the algorithm for performing an in-place turn is to set the motors to turn at the same speed in opposite directions. To generate a right turn, the left wheel should go forward and the right wheel should go backward. **Options B and C are incorrect** because they execute a swing turn to the left. **Option D is incorrect** because it executes an in-place left turn. Back to Question
32	013	B	**Option B is correct** because a collision occurs whenever the distance between the two centers is less than or equal to the sum of the two radii. **Options A, C and D are incorrect** because they do not describe the complete conditions when a collision occurs. Back to Question

Study Plan Sheet

STUDY PLAN						
Content covered on test	**How well do I know the content?**	**What material do I have for studying this content?**	**What material do I need for studying this content?**	**Where can I find the materials I need?**	**Dates planned for study of content**	**Date Completed**

Preparation Resources

The resources listed below may help you prepare for the TExES test in this field. These preparation resources have been identified by content experts in the field to provide up-to-date information that relates to the field in general. You may wish to use current issues or editions to obtain information on specific topics for study and review.

Journals

Converge, Center for Digital Education, e.Republic Inc. — www.centerdigitaled.com

Journal of Computing in Teacher Education, Taylor & Francis Group — www.tandfonline.com/toc/ujdl19/23/4

Journal of Research on Technology in Education, International Society for Technology in Education — www.iste.org

Texas Computer Education Association TechEdge and TechNotes — www.tcea.org/about/publications

The ACM Transactions on Computing Education — toce.acm.org

State-adopted Instructional Materials

tea.texas.gov/Curriculum_and_Instructional_Programs/

Other Resources

Bitter, G., and Legacy, J. (2008). *Using Technology in the Classroom*. Boston, Mass.: Allyn and Bacon.

Brookshear, J. G. (2009). *Computer Science: An Overview*. Boston, Mass.: Pearson Addison Wesley.

Campione, M., Walrath, K., and Huml, A. (2006). *The Java Tutorial: A Short Course on the Basics*. Upper Saddle River, N.J.: Pearson/Addison-Wesley.

Cormen, T. H., Leiserson, C. E., Rivest, R. L., and Stein, C. (2009). *Introduction to Algorithms*. New York, N.Y.: MIT Press, McGraw-Hill.

Goodrich, M. T., and Tamassia, R. (2010). *Data Structures and Algorithms in Java*. Indianapolis, Ind.: Wiley Text Books.

Horstmann, C. (2008). *Big Java*. Indianapolis, Ind.: Wiley Text Books.

Lever-Duffy, J., McDonald, J., and Mizell, A. P. (2008). *Teaching and Learning with Technology*. Boston, Mass.: Pearson Allyn & Bacon.

Lockard, J., and Abrams, P. (2004). *Computers for Twenty-First Century Educators*. Boston, Mass.: Pearson Allyn & Bacon.

Norton, P. (2006). *Computing Fundamentals*. Westerville, Ohio: Glencoe/McGraw-Hill.

Oualline, S. (1997). *Practical C Programmin*g. Sebastopol, Calif.: O'Reilly & Associates.

Roblyer, M. D. and Doering A. (2010). *Integrating Educational Technology into Teaching*. Boston, Mass.: Pearson Allyn & Bacon.

Sebesta, R. W. (2010). *Concepts of Programming Languages*. Boston, Mass.: Pearson Addison- Wesley.

Sommerville, I. (2006). *Software Engineering*. Boston, Mass.: Addison-Wesley Publishing Co.

Tannenbaum, A. S. (2006). *Structured Computer Organization*. Upper Saddle River, N.J.: Prentice Hall.

Weiss, M. A. (2007). *Data Structures and Algorithm Analysis in C++*. Boston, Mass.: Pearson Addison-Wesley.

Online Resources

AP Computer Science —

apcentral.collegeboard.com/apc/public/courses/teachers_corner/4483.html

Association for Computing Machinery — www.acm.org/education

Association for Computing Machinery (ACM) Special Interest Group on Computer Science Education — www.sigcse.org

Blue Pelican Java (free Java textbook and videos) — www.bluepelicanjava.com

Code.org — code.org/educate/curriculum

Code Highschool (codeHS) — codehs.com

CS Unplugged — csunplugged.org

eSchoolNews — www.eschoolnews.com/2013/12/10/computer-science-resources-168

Hour of Code — hourofcode.com

International Society for Technology in Education (ISTE) — www.iste.org

ISTE Standards for Computer Science Educators — www.iste.org/standards/iste-standards/standards-for-computer-science-educators

Javabat (free online Java interactive learning tool) — www.javabat.com

National Center for Women & Information Technology (NCWIT) — www.ncwit.org

Project Lead the Way (PLTW) — www.pltw.org/our-programs/computer-science

Project Share — projectsharetexas.org

Stanford Computer Science, Nifty Assignments from the Annual SIGCSE Meeting — nifty.stanford.edu

State Board for Educator Certification — tea.texas.gov/About_TEA/Leadership/State_Board_for_Educator_Certificatio

Texas Computer Education Association (TCEA) — www.tcea.org

Texas Computer Education Associate (TCEA) Tech-Apps/Computer Science Special Interest Group — www.tcea.org/membership/sigs/tacs-sig

Texas Education Agency, Technology Applications — tea.texas.gov/Curriculum_and_Instructional_Programs/Subject_Areas/Technology_Applications/Technology_Applications

Texas Education Agency, Texas Essential Knowledge and Skills (TEKS) — tea.texas.gov/curriculum/teks

U.S. Department of Education — www.ed.gov