STATE BOARD FOR EDUCATOR CERTIFICATION/TEXAS EDUCATION AGENCY
TEXAS EDUCATOR CERTIFICATION
TExES/ExCET TOPT TExMaT TASC/TASC-ASL

TExES | Texas Examinations of Educator Standards

Preparation Manual

141 Computer Science 8–12

PREFACE

The State Board for Educator Certification (SBEC) has developed new standards for Texas educators that delineate what the beginning educator should know and be able to do. These standards, which are based on the state-required curriculum for students—the Texas Essential Knowledge and Skills (TEKS)—form the basis for new Texas Examinations of Educator Standards (TExES™). This initiative will impact all areas of Texas education—from the more than 100 approved Texas educator preparation programs to the more than 7,000 Texas school campuses. This standards-based system reflects the SBEC's commitment to help align Texas education from kindergarten through college. The SBEC's role in this K–16 initiative will ensure that newly certified Texas teachers have the essential knowledge and skills to teach the TEKS to the state's public school students.

This manual is designed to help examinees prepare for the new TExES test in this field. Its purpose is to familiarize examinees with the competencies to be tested, test item formats, and pertinent study resources. Educator preparation program staff may also find this information useful as they help examinees prepare for careers as Texas educators.

If you have any questions after reading this preparation manual or you would like additional information about the new TExES tests or the educator standards, please visit the SBEC Web site at www.sbec.state.tx.us.

KEY FEATURES OF THE MANUAL

List of competencies that will be tested

Strategies for answering test questions

Sample test items and answer key

TABLE OF CONTENTS

SECTION I

THE NEW TExES TESTS FOR TEXAS TEACHERS

As required by the Texas Education Code §21.048, successful performance on educator certification examinations is required for the issuance of a Texas educator certificate. Each TExES test is a criterion-referenced examination designed to measure the knowledge and skills delineated in the corresponding TExES test framework. Each test framework is based on standards that were developed by Texas educators and other education stakeholders.

Each newly developed TExES test is designed to measure the requisite knowledge and skills that an entry-level educator in this field in Texas public schools must possess. The tests may include both individual, or stand-alone, test items (questions) and items that are arranged in clustered sets based on real-world situations faced by educators.

Development of the New TExES Tests

Committees of Texas educators and interested citizens guide the development of the new TExES tests by participating in each stage of the test development process. These working committees are comprised of Texas educators from public and charter schools, faculty from educator preparation programs, education service center staff, representatives from professional educator organizations, content experts, and members of the business community. The committees are balanced in terms of position, affiliation, years of experience, ethnicity, gender, and geographical location. The committee membership is rotated during the development process so that numerous Texas stakeholders may be actively involved. The steps in the process to develop the TExES tests are described below.

1. **Develop Standards.** Committees are convened to recommend what the beginning educator should know and be able to do. To ensure vertical alignment of standards across the range of instructional levels, individuals with expertise in early childhood, elementary, middle, or high school education meet jointly to articulate the critical knowledge and skills for a particular content area. Participants begin their dialogue using a "clean slate" approach with the Texas Essential Knowledge and Skills (TEKS) as the focal point. Draft standards are written to incorporate the TEKS and to expand upon that content to ensure that all beginning educators possess the appropriate level of both knowledge and skills to instruct students successfully.
2. **Review Standards.** Committees review and revise the draft standards. The revised draft standards are then placed on the SBEC Web site for public review and comment. These comments are used to prepare a final draft of the standards that will be presented to the SBEC Board for discussion, the State Board of Education (SBOE) for review and comment, and the SBEC Board for approval. Standards not based specifically on the TEKS, such as those for librarians and counselors, are proposed as rule by the SBEC Board; sent to the SBOE for its 90-day review; and, if not rejected by the SBOE, adopted by the SBEC Board.
3. **Develop Test Frameworks.** Committees review draft test frameworks that are based on the standards. These frameworks outline the specific competencies to be measured on the new TExES tests. The TExES competencies represent the critical components of the standards that can be measured with either a pencil-and-paper-based or computer-based examination, as appropriate. Draft frameworks are not finalized until after the standards are approved and the job analysis/content validation survey (see #4) is complete.

4. **Conduct Job Analysis/Content Validation Surveys.** A representative sample of Texas educators who practice in or prepare individuals for each of the fields for which an educator certificate has been proposed are surveyed to determine the relative job importance of each competency outlined in the test framework for that content area. Frameworks are revised as needed following an analysis of the survey responses.
5. **Develop and Review New Test Items.** The test contractor develops draft items that are designed to measure the competencies described in the test framework. Committees review the newly developed test items that have been written to reflect the competencies in the new test frameworks. Committee members scrutinize the draft items for appropriateness of content and difficulty; clarity; match to the competencies; and potential ethnic, gender, and regional bias.
6. **Conduct Pilot Test of New Test Items.** All of the newly developed test items that have been deemed acceptable by the item review committees are then administered to an appropriate sample of candidates for certification.
7. **Review Pilot Test Data.** Pilot test results are reviewed to ensure that the test items are valid, reliable, and free from bias.
8. **Administer New TExES Tests.** New TExES tests are constructed to reflect the competencies, and the tests are administered to candidates for certification.
9. **Set Passing Standard.** A Standard Setting Committee convenes to review performance data from the initial administration of each new TExES test and to recommend a final passing standard for that test. The SBEC considers this recommendation as it establishes a passing score on the test.

Taking the TExES Test and Receiving Scores

Please refer to the current TExES registration bulletin for information on test dates, sites, fees, registration procedures, and policies.

You will be mailed a score report approximately four weeks after each test you take. The report will indicate whether you have passed the test and will include:

- a total test *scaled* score. Scaled scores are reported to allow for the comparison of scores on the same content-area test taken on different test administration dates. The total scaled score is not the percentage of items answered correctly and is not determined by averaging the number of questions answered correctly in each domain.
  - — For all TExES tests, the score scale is 100–300 with a scaled score of 240 as the minimum passing score. This score represents the minimum level of competency required to be an entry-level educator in this field in Texas public schools.
- your performance in the major content domains of the test and in the specific content competencies of the test.
  - — This information may be useful in identifying strengths and weaknesses in your content preparation and can be used for further study or for preparing to retake the test.
- information to help you understand the score scale and interpret your results.

You will <u>not</u> receive a score report if you are absent or choose to cancel your score.

Additionally, unofficial score report information will be posted on the Internet on the score report mailing date of each test administration. Information about receiving unofficial scores via the Internet, the score scale, and other score report topics may be found on the SBEC Web site at www.sbec.state.tx.us.

Educator Standards

Complete, approved educator standards are posted on the SBEC Web site at www.sbec.state.tx.us.

SECTION II

USING THE TEST FRAMEWORK

The Texas Examination of Educator Standards (TExES) test measures the content knowledge required of an entry-level educator in this field in Texas public schools. This manual is designed to guide your preparation by helping you become familiar with the material to be covered on the test.

When preparing for this test, you should focus on the competencies and descriptive statements, which delineate the content that is eligible for testing. A portion of the content is represented in the sample items that are included in this manual. These test questions represent only a *sample* of items. Thus, your test preparation should focus on the complete content eligible for testing, as specified in the competencies and descriptive statements.

Organization of the TExES Test Framework

The test framework is based on the educator standards for this field.

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:

1. 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, and
2. the *descriptive statements,* which describe in greater detail the knowledge and skills eligible for testing.

The educator standards being assessed within each domain are listed for reference at the beginning of the test framework, which begins on page 8. These are then followed by a complete set of the framework's competencies and descriptive statements.

An example of a competency and its accompanying descriptive statements is provided on the next page.

Sample Competency and Descriptive Statements

Computer Science 8–12

Competency:

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.

Descriptive Statements:

The beginning teacher:

- Knows technology terminology and concepts.
- Demonstrates knowledge of various types of networks (e.g., LAN, WAN) and models for defining network standards and protocols (e.g., OSI, TCP/IP).
- Knows the appropriate use of hardware components (e.g., input, processing, output, primary/secondary storage devices), operating systems, software applications, and networking components.
- Knows how to select, connect, and use a variety of input, output, and storage devices and peripherals (e.g., scanner, voice/sound recorders, touch screen, digital camera, printer).
- Knows how to evaluate software (e.g., graphics, animation, multimedia, video, Web authoring) for quality, appropriateness, effectiveness, and efficiency and how to make decisions regarding its proper acquisition and use.
- Knows how to perform basic application functions (e.g., opening an application program; creating, modifying, saving, and printing documents) and how to access, manage, and manipulate information from secondary storage devices.
- Knows strategies for acquiring information from electronic resources (e.g., encyclopedias, databases, libraries of images, reference software, Internet).
- Knows search strategies (e.g., keyword, Boolean, natural language) for locating and retrieving information in electronic formats (e.g., text, audio, video, graphics).
- Knows how to assess the accuracy and validity of acquired information.
- Knows how to resolve information conflicts through research and comparison of data from multiple sources.
- Demonstrates knowledge of the ethical acquisition (e.g., citing sources using established methods) and acceptable versus unacceptable use of information (e.g., privacy, hacking, piracy, vandalism, viruses, current laws and regulations).
- Demonstrates knowledge of intellectual property rights and related issues (e.g., copyright laws, fair use, patents, trademarks) when using, manipulating, and editing electronic data.
- Knows how to use online help and other support documentation.
- Knows how to use technical-writing strategies to develop documentation for a variety of communication products.

- Demonstrates knowledge of the impact of technology on society and the importance of technology to future careers, lifelong learning, and daily living for individuals of all ages.
- Investigates measures (e.g., passwords, virus detection/prevention) to protect computer systems and databases from unauthorized use and tampering.

Studying for the TExES Test

The following steps may be helpful in preparing for the TExES test.

1. Identify the information the test will cover by reading through the test competencies (see the following pages in this section). *Within each domain* of this TExES test, each competency will receive approximately equal coverage.

2. Read each competency with its descriptive statements in order to get a more specific idea of the knowledge you will be required to demonstrate on the test. You may wish to use this review of the competencies to set priorities for your study time.

3. Review the "Preparation Resources" section of this manual for possible resources to consult. Also, compile key materials from your preparation coursework that are aligned with the competencies.

4. Study this manual for approaches to taking the TExES test.

5. When using resources, concentrate on the key ideas and important concepts that are discussed in the competencies and descriptive statements.

NOTE: This preparation manual is the only TExES test study material endorsed by the SBEC for this field. Other preparation materials may not accurately reflect the content of the test or the policies and procedures of the TExES program.

TEST FRAMEWORK FOR FIELD 141: COMPUTER SCIENCE 8–12

Domain I **Technology Applications Core (approximately 33.3% of the test)**

Standards Assessed:

Computer Science 8–12 Standard I:
All teachers use technology-related terms, concepts, data input strategies, and ethical practices to make informed decisions about current technologies and their applications.

Computer Science 8–12 Standard II:
All teachers identify task requirements, apply search strategies, and use current technology to efficiently acquire, analyze, and evaluate a variety of electronic information.

Computer Science 8–12 Standard III:
All teachers use task-appropriate tools to synthesize knowledge, create and modify solutions, and evaluate results in a way that supports the work of individuals and groups in problem-solving situations.

Computer Science 8–12 Standard IV:
All teachers communicate information in different formats and for diverse audiences.

Computer Science 8–12 Standard V:
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 VI:
The computer science teacher has the knowledge and skills needed to teach the Foundations, Information Acquisition, Work in Solving Problems, and Communication 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–V.

Domain II **Program Design and Development (approximately 33.3% of the test)**

Standards Assessed:

Computer Science 8–12 Standard VI:
The computer science teacher has the knowledge and skills needed to teach the Foundations, Information Acquisition, Work in Solving Problems, and Communication 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–V.

Domain III Programming Language Topics (approximately 33.3% of the test)

Standards Assessed:

Computer Science 8–12 Standard VI:
The computer science teacher has the knowledge and skills needed to teach the Foundations, Information Acquisition, Work in Solving Problems, and Communication 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–V.

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:

- Knows technology terminology and concepts.
- Demonstrates knowledge of various types of networks (e.g., LAN, WAN) and models for defining network standards and protocols (e.g., OSI, TCP/IP).
- Knows the appropriate use of hardware components (e.g., input, processing, output, primary/secondary storage devices), operating systems, software applications, and networking components.
- Knows how to select, connect, and use a variety of input, output, and storage devices and peripherals (e.g., scanner, voice/sound recorders, touch screen, digital camera, printer).
- Knows how to evaluate software (e.g., graphics, animation, multimedia, video, Web authoring) for quality, appropriateness, effectiveness, and efficiency and how to make decisions regarding its proper acquisition and use.
- Knows how to perform basic application functions (e.g., opening an application program; creating, modifying, saving, and printing documents) and how to access, manage, and manipulate information from secondary storage devices.
- Knows strategies for acquiring information from electronic resources (e.g., encyclopedias, databases, libraries of images, reference software, Internet).
- Knows search strategies (e.g., keyword, Boolean, natural language) for locating and retrieving information in electronic formats (e.g., text, audio, video, graphics).
- Knows how to assess the accuracy and validity of acquired information.
- Knows how to resolve information conflicts through research and comparison of data from multiple sources.
- Demonstrates knowledge of the ethical acquisition (e.g., citing sources using established methods) and acceptable versus unacceptable use of information (e.g., privacy, hacking, piracy, vandalism, viruses, current laws and regulations).
- Demonstrates knowledge of intellectual property rights and related issues (e.g., copyright laws, fair use, patents, trademarks) when using, manipulating, and editing electronic data.
- Knows how to use online help and other support documentation.
- Knows how to use technical-writing strategies to develop documentation for a variety of communication products.
- Demonstrates knowledge of the impact of technology on society and the importance of technology to future careers, lifelong learning, and daily living for individuals of all ages.
- Investigates measures (e.g., passwords, virus detection/prevention) to protect computer systems and databases from unauthorized use and tampering.

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:

- Knows how to plan, create, and edit documents using word processing features (e.g., readable fonts, alignment, page setup, tabs, ruler settings) to solve problems and communicate results.
- Knows how to plan, create, and edit spreadsheets using spreadsheet features (e.g., data types, formulas, functions, charts) to solve problems and communicate results.
- Knows how to plan, create, and edit databases using database features (e.g., defining fields, entering data, creating horizontal and vertical layouts) to solve problems and communicate results.
- Knows how to integrate one or more objects (e.g., tables, charts, graphs, graphics) into a product.
- Knows how to use productivity tools to create products (e.g., slide shows, posters, multimedia presentations, spreadsheets) for defined audiences.
- Knows how to publish information in a variety of ways (e.g., printed copy, monitor displays, Internet documents, video).
- Knows how to use telecommunications tools (e.g., Internet browsers, video conferencing, distance learning) for a variety of purposes.
- Knows how to use interactive virtual environments (e.g., virtual field trips, instructional simulations).
- Knows how to use collaborative software.
- Knows how to share information through online communication.
- Demonstrates knowledge of issues concerning proper etiquette when communicating using electronic tools.
- Demonstrates knowledge of how to design and implement procedures to track trends, set timelines, and review and evaluate products using technology tools (e.g., database managers, daily/monthly planners, project management tools).
- Knows how to evaluate projects for design, purpose, audience, and content delivery using various criteria (e.g., technology specifications, established criteria, rubrics).
- Knows how to select representative products to be collected and stored in an electronic evaluation tool and how to evaluate products for relevance to the assignment or task.
- Knows how to plan and design communication products that are accessible to learners with diverse needs and abilities.

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) for all students.

The beginning teacher:

- Knows how to plan computer science lessons using a range of instructional strategies for individuals and groups.
- Demonstrates knowledge of issues related to the equitable use of technology (e.g., gender, ethnicity, language, disabilities, access to technology).
- Knows how to plan and implement instruction that allows students to use computer science in problem-solving and decision-making situations.
- Knows how to develop and facilitate collaborative tasks and teamwork among group members.
- Knows how to use technology tools to perform administrative tasks (e.g., attendance, grades, communication).
- Knows how to use a variety of instructional strategies to ensure students' reading comprehension.
- Knows strategies to help students learn how to locate, retrieve, analyze, evaluate, communicate, and retain content-related information.
- Knows how to evaluate student projects and portfolios using formal and informal assessment methods.
- Knows the relationship between instruction and assessment and uses assessment results for gauging student progress and adjusting instruction.
- Identifies resources to keep current with the use of technology in education and issues related to legal and ethical use of technology resources.
- Knows how to use technology to participate in self-directed activities in society and how to participate within electronic communities in a variety of roles (e.g., collaborator, learner, contributor, teacher/mentor).

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:

- Exhibits knowledge of the analysis and design phases of the software system life cycle.
- Knows the characteristics of programming design strategies.
- Knows how to apply problem-solving strategies (e.g., design specification, top-down design, step-wise refinement, object-oriented design).
- Demonstrates the ability to compare and contrast design strategies (e.g., top-down, bottom-up, object-oriented).
- Demonstrates the use of visual organizers (e.g., flowcharts, schematic drawings) to design solutions to problems.
- 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:

- Knows the characteristics of models (e.g., waterfall, incremental, spiral) used in the development of software systems.
- Knows how to survey the issues accompanying the development of large software systems (e.g., design/implementation teams, software validation/ testing, risk assessment).
- Demonstrates the use of programming style conventions (e.g., spacing, indentation, descriptive identifiers, comments, documentation) to enhance the readability and functionality of code.
- Knows how to create robust programs with emphasis on style, clarity of expression, and documentation to facilitate maintenance, program expansion, reliability, validity, and efficiency.
- Knows how to create and use libraries of generic modular code to be used for efficient programming.
- Demonstrates the ability to read and modify large programs, including design description and process development.
- Demonstrates effective use of predefined input and output, including logic to protect from invalid input.
- Demonstrates the ability to debug and solve problems using reference materials and effective strategies.
- Knows how to determine and employ methods to evaluate the design and functionality of information acquisition processes and algorithms, using effective coding, design, and test data.

Competency 006

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

The beginning teacher:

- Knows necessary vocabulary related to computer science (e.g., cache, bits, encryption).
- Knows specific programming terminology (e.g., data type, data structure, encapsulation) and programming concepts (e.g., procedural, object-oriented).
- Demonstrates knowledge of advanced computer science concepts (e.g., computer architecture, operating systems, artificial intelligence).
- Demonstrates the ability to use notation for language definition (e.g., syntax diagrams, Backus-Naur forms).
- Knows the differences in the levels of languages (e.g., machine, assembly, high-level compiled, interpreted).
- Knows the characteristics of and differences in current programming languages and paradigms.
- Demonstrates knowledge of the uses of current programming languages and paradigms in other fields of study.

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:

- Knows the characteristics and uses of constants, variables, and simple data types in current programming languages (e.g., int, short, char, double, boolean).
- Demonstrates effective use of standard and user-defined methods or functions in the development of code.
- Knows how to identify and use parameters, both actual and formal, and how to pass parameters by value and by reference.
- Knows how to identify object-oriented data types and delineate the advantages and disadvantages of object data.
- Demonstrates the ability to identify and use one-dimensional arrays, records, and sequential and nonsequential files.
- Knows how to identify and use multidimensional arrays and arrays of records.
- Demonstrates the ability to develop coding with the use of data structures, and to manipulate data structures using string processing routines (e.g., concatenation of strings, substring search).
- Knows the characteristics of and develops code using 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 knows how to:

- Apply standard operators (e.g., arithmetic, relational, logical, assignment, increment/decrement, input/output) and correct operator precedence.
- Identify the characteristics of control structures.
- Use conditional control structures (e.g., if, if . . . else statements).
- Construct iterative control structures (e.g., for and while statements, do loops).
- Use pretest (e.g., for, while) and posttest (e.g., do . . . while) loops.
- Use sequential, conditional, selection, and repetition execution control structures such as menu-driven programs that branch and allow user input.
- Demonstrate coding proficiency in contemporary programming languages, including an object-oriented language.

Competency 009

The computer science teacher knows how to construct, compare, and analyze various algorithms.

The beginning teacher knows how to:

- Construct searching algorithms (e.g., linear and binary searches).
- Construct sorting algorithms (e.g., selection, bubble, insertion, merge, shell, and quick sorts).
- Compare and contrast searching and sorting algorithms for space and time requirements.
- Construct and appropriately use iterative and recursive algorithms.
- Compare and contrast iterative and recursive algorithms.
- Develop sequential, iterative, and recursive algorithms and code programs in prevailing computer languages to solve practical problems.
- Analyze various algorithms using “big-O” notation and best-, average-, and worst-case space and time techniques.
- Identify and describe the correctness and complexity of specific types of algorithms (e.g., divide and conquer, greedy, backtracking).

SECTION III

APPROACHES TO ANSWERING MULTIPLE-CHOICE ITEMS

The purpose of this section is to describe multiple-choice item formats that you may see on the TExES test in this field and to suggest possible ways to approach thinking about and answering the multiple-choice items. However, these approaches are not intended to replace familiar test-taking strategies with which you are already comfortable and that work for you.

The Computer Science 8–12 test is designed to include 80 scorable multiple-choice items and approximately 10 nonscorable items. Your final scaled score will be based only on scorable items. The nonscorable multiple-choice items are pilot tested by including them in the test in order to collect information about how these questions will perform under actual testing conditions. Nonscorable test items are not considered in calculating your score, and they are not identified on the test.

All multiple-choice questions on this test are designed to assess your knowledge of the content described in the test framework. The multiple-choice questions assess your ability to recall factual information **and** to think critically about the information, analyze it, consider it carefully, compare it with other knowledge you have, or make a judgment about it.

When you are ready to answer a multiple-choice question, you must choose one of four *answer choices* labeled A, B, C, and D. Then you must mark your choice on a separate answer sheet.

Item Formats

You may see the following two types of multiple-choice questions on the test.

- Single items
- Items with stimulus material

You may have two or more items related to a single stimulus. This group of items is called a cluster. Following the last item of a clustered item set containing two or more items, you will see the graphic illustrated below.

◆◆◆◆◆◆◆◆

This graphic is used to separate these clustered items related to specific stimulus material from other items that follow.

On the following pages, you will find descriptions of these commonly used item formats, along with suggested approaches for answering each type of item. In the actual testing situation, you may mark the test items and/or write in the margins of your test booklet, **but your final response must be indicated on the answer sheet provided.**

SINGLE ITEMS

In the single item format, a problem is presented as a direct question or an incomplete statement, and four answer choices appear below the question. The following question is an example of this type. 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) for all students.*

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 programming language.

B. require the students to read several books written on the language.

C. give the students quizzes on a variety of topics pertaining 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 it on your answer sheet.

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 response options and consider which of them 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 to 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 to 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 pertaining to a programming language is the most effective way of helping them to 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 to 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 to improve their comprehension of its concepts and constructs. By writing programs and then testing them, students are able to implement various concepts of a programming language and, in doing so, can improve their understanding of these 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 to improve their comprehension of the language's concepts and constructs. Therefore, the correct response is option D.

In the single item format, a problem is presented as a direct question or an incomplete statement, and four answer choices appear below the question. The following question is an example of this type. It tests knowledge of Computer Science 8–12 competency 004: *The computer science teacher knows problem-solving strategies and different procedures for program design.*

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 it on your answer sheet.

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 options and consider which of them 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, such as 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, such as 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, such as 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, such as 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.

ITEMS WITH STIMULUS MATERIAL

Some questions are preceded by stimulus material that relates to the item. Some types of stimulus material included on the test are reading passages, 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.

One or more items may be related to a single stimulus. You can use several different approaches to answer these types of questions. Some commonly used approaches are listed below.

Strategy 1 Skim the stimulus material to understand its purpose, its arrangement, and/or its content. Then read the item and refer again to the stimulus material to verify the correct answer.

Strategy 2 Read the item *before* considering the stimulus material. The content of the item will help you identify the purpose of the stimulus material and locate the information you need to answer the question.

Strategy 3 Use a combination of both strategies; apply the "read the stimulus first" strategy with shorter, more familiar stimuli and the "read the item first" strategy with longer, more complex, or less familiar stimuli. You can experiment with the sample items 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 item, you should read it carefully and critically. You may want to underline its important points to help you answer the item.

As you consider items set in educational contexts, try to use that teacher's point of view to answer the items that accompany the stimulus. Be sure to consider the items in terms of only the information provided in the stimulus—not in terms of specific situations or individuals you may have encountered.

Suggested Approach

First read the stimulus (a segment of code).

Use the code segment below to answer the question that follows.

```
1.  k = 18
2.  for j = 1 to 5
3.  {
4.     if  (k > 10)
5.
6.  }
```

Now you are prepared to address the question associated with this stimulus. The question measures 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.*

If the code above is executed and the resulting value of k is 10, which of the following statements must have been on line 5?

A. $k = k - 1$

B. $k = k - 3$

C. $k = k - 4$

D. $k = k - 6$

Suggested Approach

Examine carefully the code segment presented in the stimulus; then read the question. Now consider the response options to determine which one would have to be on line 5 of the code segment so that, after the code is executed, it yields a value of 10 for the variable *k*. The initial value of *k* is 18 and the for statement initializes the variable *j* to 1, increments it by 1, and terminates it at 5.

Option A, which indicates that $k = k - 1$ would be on line 5 of the code segment, would result in a value of 13 for *k*. Thus, option A is an incorrect response.

Option B, which indicates that $k = k - 3$ would be on line 5 of the code segment, would result in a value of 9 for *k*. Thus, option B is an incorrect response.

Option C, which indicates that $k = k - 4$ would be on line 5 of the code segment, would result in a value of 10 for *k*. Thus, option C is a correct response.

Option D, which indicates that $k = k - 6$ would be on line 5 of the code segment, would result in a value of 6 for *k*. Thus, option D is an incorrect response.

Of the alternatives offered, only putting $k = k - 4$ on line 5 of the code segment would result in a value of 10 for *k*. Therefore, the correct response is option C.

SECTION IV

SAMPLE ITEMS

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

An answer key follows the sample items. The answer key lists the item number and correct answer for each sample test item. Please note that the answer key also lists the competency assessed by each item and that the sample items are not necessarily presented in competency order.

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

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 by 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 e-mail attachment.

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

Competency 002

3. Students in a Texas classroom have been communicating with a class in New York by video conference. 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.

Competency 002

4. 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. GIF

   D. HTML

Competency 003

5. 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 e-mail messages with attached document files.

   C. update pages on the class's Web site.

   D. engage in dialogue in a real-time chat room.

Competency 004

6. 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

Competency 004

7. Which of the following would best facilitate the expansion of a computer program?

   A. incorporation of diagrams in the design documents

   B. minimal use of global variables in the source code

   C. construction of methods that are highly dependent on one another

   D. extensive use of functions that support the debugging process

Competency 005

8. A software system is to be developed for a customer who requires emphasis on risk assessment. To meet this customer's requirements, which of the following models would be most appropriate to use?

   A. chaos

   B. incremental

   C. spiral

   D. waterfall

Competency 005

9. 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.

Competency 006

10. 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

Competency 006

11. A multibyte data representation is stored in memory with its most significant byte in the lowest memory address. Which of the following describes this method of addressing?

    A. ASCII

    B. big-endian

    C. NUXI ordering

    D. Huffman encoding

Competency 007

12. 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

Competency 007

13. A heap is best represented using which of the following?

    A. a binary tree

    B. a graph

    C. a linked list

    D. a stack

Competency 008

14. **Use the code segments below to answer the question that follows.**

Func1(*v*)
{
  t = 0
  do
  {
  t = *t* + 1
  print *t*
  }
  while (*t* < *v*)
}
	Func2(*v*)
	{
	  t = 0
	  while (*t* < *v*)
	  {
	  t = *t* + 1
	  print *t*
	  }
	}

The functions *Func1* and *Func2* above will print the same values of *t* for which of the following values of *v*?

A. $v > 0$

B. $v \geq 0$

C. $v < 0$

D. $v \leq 0$

Competency 009

15. Which of the following represents the average-case performance of a quick sort algorithm?

A. $O(n)$

B. $O(\log_2 n)$

C. $O(n^2)$

D. $O(n \log_2 n)$

Competency 009

16. **Use the code segment below to answer the question that follows.**

```
Rep(n)
{
  if (n >= 4)
    Rep(n / 2)
  print n
}
```

Which of the following printouts results from calling the function *Rep*(16)?

A. 2

B. 16

C. 16
8
4
2

D. 2
4
8
16

Competency 009

17. A specific sorting algorithm involves repeatedly comparing two adjacent elements in a list and subsequently interchanging them if they are not in the correct order relative to one another. Which of the following is the sorting algorithm characterized by this description?

A. quick sort

B. bubble sort

C. heap sort

D. selection sort

ANSWER KEY

Item Number	Correct Answer	Competency
1	**B**	001
2	**C**	001
3	**A**	002
4	**C**	002
5	**A**	003
6	**B**	004
7	**B**	004
8	**C**	005
9	**A**	005
10	**C**	006
11	**B**	006
12	**C**	007
13	**A**	007
14	**A**	008
15	**D**	009
16	**D**	009
17	**B**	009

SECTION V

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

Connection Science, Taylor & Francis Group.

Converge Online, Center for Digital Education, e.Rupblic Inc., http://www.centerdigitaled.com/converge

Journal of Computing in Teacher Education, International Society for Technology in Education, http://www.iste.org

Journal of Research on Technology in Education, International Society for Technology in Education, http://www.iste.org

Journal on Educational Resources in Computing, Association for Computing Machinery.

Texas Computer Education Association TechEdge and TechNotes, http://www.tcea.org

Other Sources

Bitter, G., and Pierson, M. (2004). *Using Technology in the Classroom.* Boston, MA: Allyn and Bacon.

Brookshear, J. G. (2002). *Computer Science: An Overview*. Boston, MA: Pearson Addison Wesley.

Campione, M., Walrath, K., and Huml, A. (2001). *The Java Tutorial: A Short Course on the Basics.* Upper Saddle River, NJ: Pearson/Addison-Wesley

Cormen, T. H., Leiserson, C.E., Rivest, R.L., and Stein, C. (2001). *Introduction to Algorithms.* New York, NY: MIT Press, McGraw-Hill.

Goodrich, M. T., and Tamassia, R. (2003). *Data Structures and Algorithms in Java*. Indianapolis, IN: Wiley Text Books.

Knowlton, T. (1998). *Introduction to Computer Science Using C++.* Cincinnati, OH: South-Western Educational Publishing.

Lambert, K. A., and Nance, D. W. (1997). *Fundamentals of C++: Understanding Programming and Problem Solving.* Boston, MA: Course Technology, Thomson Learning.

Lambert, K. A., and Naps, T.L. (1998). *Fundamentals of Program Design and Data Structures with C++.* Cincinnati, OH: South-Western Educational Publishing.

Lever-Duffy, J., and Mizell, A. P. (2002). *Teaching and Learning with Technology.* Boston, MA: Pearson Allyn & Bacon.

Lockard, J., and Abrams, P. (2003). *Computers for Twenty-First Century Educators.* Boston, MA: Pearson Allyn & Bacon.

Norton, P. (1999). *Computing Fundamentals.* Westerville, OH: Glencoe/McGraw-Hill.

Oualline, S. (1991). *Practical C Programming.* Sebastopol, CA: O'Reilly & Associates.

Roblyer, M. D. (2004). *Integrating Educational Technology into Teaching.* Upper Saddle River, NJ: Prentice Hall.

Sebesta, R. W. (2003). *Concepts of Programming Languages*. Boston, MA: Pearson Addison-Wesley.

Sedgewick, R. (1998). *Algorithms in C++: Parts 1–4: Fundamentals, Data Structures, Sorting, Searching.* Boston, MA: Addison-Wesley.

Sommerville, I. (2000). *Software Engineering.* Boston, MA: Addison-Wesley Publishing Co.

Tannenbaum, A. S. (1998). *Structured Computer Organization.* Upper Saddle River, NJ: Prentice Hall.

Online Resources

Association for Computing Machinery, http://www.acm.org

Association for Computing Machinery (ACM) Special Interest Group on Computer Science Education, http://www.sigcse.org

Computer Science Teaching Center, http://www.cstc.org/index.html

State Board for Educator Certification (Standards & Testing), http://www.sbec.state.tx.us

State Board of Education Certification, Standards and Testing, Educator Standards, http://www.sbec.state.tx.us/SBECOnline/standtest/educstan.asp

Technology Applications Teacher Network, http://www.techappsnetwork.org

Texas Computer Education Association (TCEA), http://www.tcea.org

Texas Computer Education Associate (TCEA) Tech-Apps/Computer Science Special Interest Group, http://www.tcea.org/SiGs

Texas Education Agency, Educational Technology, http://www.tea.state.tx.us/curriculum

Texas Education Agency, Technology Applications Curriculum, http://www.tea.state.tx.us/curriculum

Texas Education Agency, Texas Essential Knowledge and Skills (TEKS), http://www.tea.state.tx.us/teks

U.S. Department of Education, http://www.ed.gov

State-Adopted Instructional Materials

http://www.tea.state.tx.us/textbooks