Welcome to The Praxis® Study Companion

Prepare to Show What You Know

You have been working to acquire the knowledge and skills you need for your teaching career. Now you are ready to demonstrate your abilities by taking a Praxis® test.

Using the Praxis® Study Companion is a smart way to prepare for the test so you can do your best on test day. This guide can help keep you on track and make the most efficient use of your study time.

The Study Companion contains practical information and helpful tools, including:

- An overview of the Praxis tests
- Specific information on the Praxis test you are taking
- A template study plan
- Study topics
- Practice questions and explanations of correct answers
- Test-taking tips and strategies
- Frequently asked questions
- Links to more detailed information

So where should you start? Begin by reviewing this guide in its entirety and note those sections that you need to revisit. Then you can create your own personalized study plan and schedule based on your individual needs and how much time you have before test day.

Keep in mind that study habits are individual. There are many different ways to successfully prepare for your test. Some people study better on their own, while others prefer a group dynamic. You may have more energy early in the day, but another test taker may concentrate better in the evening. So use this guide to develop the approach that works best for you.

Your teaching career begins with preparation. Good luck!

Know What to Expect

Which tests should I take?

Each state or agency that uses the Praxis tests sets its own requirements for which test or tests you must take for the teaching area you wish to pursue.

Before you register for a test, confirm your state or agency’s testing requirements at www.ets.org/praxis/states.

How are the Praxis tests given?

Praxis tests are given on computer. Other formats are available for test takers approved for accommodations (see page 52).
What should I expect when taking the test on computer?

When taking the test on computer, you can expect to be asked to provide proper identification at the test center. Once admitted, you will be given the opportunity to learn how the computer interface works (how to answer questions, how to skip questions, how to go back to questions you skipped, etc.) before the testing time begins. Watch the What to Expect on Test Day video to see what the experience is like.

Where and when are the Praxis tests offered?

You can select the test center that is most convenient for you. The Praxis tests are administered through an international network of test centers, which includes Prometric® Testing Centers, some universities, and other locations throughout the world.

Testing schedules may differ, so see the Praxis web site for more detailed test registration information at www.ets.org/praxis/register.
# Table of Contents

*The Praxis® Study Companion guides you through the steps to success*

1. **Learn About Your Test** ................................................................. 5  
   Learn about the specific test you will be taking

2. **Familiarize Yourself with Test Questions** ................................. 15  
   Become comfortable with the types of questions you’ll find on the Praxis tests

3. **Practice with Sample Test Questions** ........................................ 19  
   Answer practice questions and find explanations for correct answers

4. **Determine Your Strategy for Success** ......................................... 32  
   Set clear goals and deadlines so your test preparation is focused and efficient

5. **Develop Your Study Plan** ............................................................ 35  
   Develop a personalized study plan and schedule

6. **Review Study Topics** ................................................................. 39  
   Detailed study topics with questions for discussion

7. **Review Smart Tips for Success** .................................................. 50  
   Follow test-taking tips developed by experts

8. **Check on Testing Accommodations** .......................................... 52  
   See if you qualify for accommodations to take the Praxis test

9. **Do Your Best on Test Day** .......................................................... 53  
   Get ready for test day so you will be calm and confident

10. **Understand Your Scores** ............................................................ 55  
    Understand how tests are scored and how to interpret your test scores

**Appendix: Other Questions You May Have** ...................................... 57
1. Learn About Your Test

Learn about the specific test you will be taking

Computer Science (5652)

<table>
<thead>
<tr>
<th>Test at a Glance</th>
</tr>
</thead>
<tbody>
<tr>
<td>Test Name</td>
</tr>
<tr>
<td>Test Code</td>
</tr>
<tr>
<td>Time</td>
</tr>
<tr>
<td>Number of Questions</td>
</tr>
<tr>
<td>Format</td>
</tr>
<tr>
<td>Test Delivery</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Content Categories</th>
<th>Approximate Number of Questions</th>
<th>Approximate Percentage of Examination</th>
</tr>
</thead>
<tbody>
<tr>
<td>I. Impacts of Computing</td>
<td>15</td>
<td>15%</td>
</tr>
<tr>
<td>II. Algorithms and Computational Thinking</td>
<td>25</td>
<td>25%</td>
</tr>
<tr>
<td>III. Programming</td>
<td>30</td>
<td>30%</td>
</tr>
<tr>
<td>IV. Data</td>
<td>15</td>
<td>15%</td>
</tr>
<tr>
<td>V. Computing Systems and Networks</td>
<td>15</td>
<td>15%</td>
</tr>
</tbody>
</table>

About This Test

The Praxis Computer Science test is designed to assess the computer science knowledge and competencies necessary for a beginning teacher of secondary school computer science. Examinees have typically completed a bachelor's program with an emphasis in computer science or computer science education.

The examinee will be required to understand and work with computer science concepts, use algorithms and computational thinking, work with code, manipulate data, and demonstrate knowledge of computing systems and networks.

The test is not designed to be aligned with any particular computer science curriculum, but it is intended to be consistent with the recommendations of national studies on computer science education, such as the K-12 Computer Science Framework (2016), the Computer Science Teachers Association (CSTA) K-12 Computer Science Standards (2017), and the International Society for Technology in Education (ISTE) Standards for Computer Science Educators (2011).

This test may contain some questions that do not count toward your score.
Test Specifications

Test specifications in this chapter describe the knowledge and skills measured by the test. Study topics to help you prepare to answer test questions can be found in page 39.

I. Impacts of Computing

A. Understands and applies knowledge of impact of, obstacles to, and effects of computing
   1. Understand computing as a way of expressing creativity, solving problems, enabling communication, and fostering innovation in a variety of fields and careers
      a. recognize that computers can be used to showcase creativity
      b. recognize the benefits of using computers to solve problems
      c. provide examples of how computers enable communication and collaboration
      d. provide examples of how computers foster innovation
   2. Know the obstacles to equal access to computing among different groups and the impact of those obstacles
      a. identify obstacles to equal access to computing among different groups (e.g., groups defined by gender, socioeconomic status, disability/accessibility needs) and the impact of those obstacles
      b. identify factors that contribute to the digital divide
      c. match obstacles to equal access with effective solutions
   3. Understand beneficial and harmful effects of computing innovations and the trade-offs between them
      a. analyze computing innovations in terms of their social, economic, and cultural impacts, both beneficial and harmful
      b. identify trade-offs between beneficial and harmful effects of computer innovations

B. Understands and applies knowledge of issues regarding intellectual property, ethics, privacy, and security in computing
   1. Know different methods of protecting intellectual property rights and the trade-offs between them in a variety of contexts (e.g., Creative Commons, open source, copyright)
      a. using correct vocabulary, describe how different methods of protecting intellectual property rights work
      b. given a context, identify appropriate methods of protecting intellectual property rights
      c. identify and compare trade-offs between different methods of protecting intellectual property rights
   2. Understand ethical and unethical computing practices and their social, economic, and cultural implications
      a. identify ethical and unethical computing practices in context
      b. describe the social, economic, and cultural implications of ethical and unethical computing practices
      c. identify the conditions under which a given computing practice is ethical or legal
   3. Know privacy and security issues regarding the acquisition, use, and disclosure of information in a digital world
      a. using correct vocabulary, describe privacy and security issues
      b. in context, identify appropriate strategies to safeguard privacy and ensure security
      c. describe trade-offs between local and cloud-based data storage
      d. identify methods that digital services use to collect information about users
II. Algorithms and Computational Thinking

A. Understands and applies knowledge of abstraction, pattern recognition, problem decomposition, number base conversion, and algorithm formats

1. Understand abstraction as a foundation of computer science
   a. identify, create, or complete the correct ordering, from low to high, of an abstraction hierarchy
   b. identify abstractions in context
   c. identify details that can be removed from a solution in order to generalize it

2. Know how to use pattern recognition, problem decomposition, and abstraction to develop an algorithm
   a. given a table of values or other data source, identify the patterns in the data and identify algorithms that could produce the patterns
   b. identify components that could be part of an algorithm to solve a problem
   c. identify actions and actors when decomposing a problem
   d. identify appropriate decomposition strategies

3. Understand number base conversion and binary, decimal, and hexadecimal number systems
   a. convert between number bases
   b. analyze and compare representations of numbers in different bases

4. Understand how to develop and analyze algorithms expressed in multiple formats (e.g., natural language, flowcharts, pseudocode)
   a. interpret diagrams that describe algorithms, given an explanation of the symbols used
   b. compare algorithms written in multiple formats
   c. trace and analyze algorithms written in different formats
   d. identify correct sequencing of steps in an algorithm and errors in sequencing

B. Understands and applies knowledge of algorithm analysis, searching and sorting algorithms, recursive algorithms, and randomization

1. Be familiar with the limitations of computing in terms of time, space, and solvability as well as with the use of heuristic solutions that can address these limitations
   a. identify and compare algorithms that are linear, quadratic, exponential, or logarithmic
   b. recognize the existence of problems that cannot be solved by a computer
   c. in context, identify factors that prevent a problem from being solvable
   d. identify situations where heuristic solutions are useful
   e. in context, identify space and time limitations of computational solutions to problems

2. Understand searching and sorting algorithms; can analyze sorting algorithms for correctness and can analyze searching algorithms for correctness and efficiency
   a. trace algorithms and predict output and intermediate results
   b. calculate the number of comparisons required for linear and binary search algorithms

3. Understand simple recursive algorithms (e.g., \( n! \) factorial, sum of first \( n \) integers)
   a. trace simple recursive algorithms
   b. provide missing steps in incomplete simple recursive algorithms
   c. identify parts of a recursive algorithm (e.g., base or stopping condition, recursive call)
   d. identify errors in simple recursive algorithms
   e. identify an iterative algorithm that is equivalent to a recursive algorithm

4. Be familiar with the use of randomization in computing
   a. identify appropriate uses of randomization in a variety of applications
   b. identify the difference between random and pseudorandom numbers
III. Programming

A. Understands and applies knowledge of programming control structures, standard operators, variables, correctness, extensibility, modifiability, and reusability

1. Understand how to write and modify computer programs in a text-based programming language
   a. describe what a program does or be able to choose the code segment that correctly implements a given intended purpose
   b. identify missing code in a code segment with a stated intended purpose
   c. place statements in appropriate order to create a correct program
   d. identify how changing one part of a code segment will affect the output

2. Understand how to analyze computer programs in terms of correctness
   a. trace code and indicate the output printed or the value of variables after code segment execution
   b. indicate inputs that produce given outputs for a code segment
   c. describe what a program does or choose the code segment that correctly implements a given intended purpose
   d. identify valid preconditions and postconditions
   e. compare two code segments or algorithms
   f. identify the type of error produced by a code segment (i.e., syntax, runtime, compile-time, overflow, round-off, logic)
   g. identify errors in incorrect code and changes that can be made to correct them

3. Know the concepts of extensibility, modifiability, and reusability
   a. identify the meaning of the terms
   b. identify functionally equivalent statements or code segments that differ in one of these three ways
   c. identify situations where the use of constants or variables would be preferred over hard-coded values
   d. identify opportunities for parameterization
   e. choose code that improves on given code by making it more extensible, modifiable, or reusable
   f. identify changes that would improve a given code segment

4. Understand the three basic constructs used in programming: sequence, selection, and iteration
   a. trace code and indicate the output printed or the value of variables after code segment execution
   b. indicate inputs that produce given outputs for a code segment
   c. describe what a program does or choose the code segment that correctly implements a given intended purpose
   d. identify missing code in a code segment with a stated intended purpose
   e. identify equivalent statements or code segments
   f. identify the three constructs when used in code
   g. identify which of the constructs are needed to implement given functionality
   h. convert code that does not use iteration to equivalent code that uses iteration

5. Understand how to use standard operators (i.e., assignment, arithmetic, relational, logical) and operator precedence to write programs
   a. trace code and indicate the output displayed or the value of variables after code segment execution
   b. indicate inputs that produce given outputs for a code segment
   c. describe what a program does or choose the code segment that correctly implements a given intended purpose
   d. identify missing code in a code segment with a stated intended purpose
   e. identify equivalent statements or code segments
   f. place statements in appropriate order to create a correct program
   g. use Boolean algebra to identify equivalent Boolean expressions
   h. write a Boolean expression equivalent to given code, or identify code equivalent to a given Boolean expression or English description
   i. identify the correct implementation of a given formula, including formulas with fractions
Step 1: Learn About Your Test

j. evaluate expressions that include arithmetic operations
6. Understand how to use variables and a variety of data types
   a. identify variables and data types (e.g., integers, floating point, string, Booleans, arrays/lists)
   b. identify the need for type conversion
   c. trace code and indicate the output printed or the value of variables after code segment execution
   d. indicate the inputs that produce given outputs for a code segment
   e. describe what a program does or choose the code segment that correctly implements a stated intended purpose
   f. identify missing code in a code segment with a stated intended purpose
   g. identify equivalent statements or code segments
   h. place statements in appropriate order to create a correct program
   i. describe the difference between integer and floating point numeric data types
   j. describe the difference between integer and floating point division
   k. describe the benefits of the use of each data type
   l. distinguish between global and local scope
   m. identify the most appropriate data type in a given context
   n. identify the correct sequence of string operations to produce a given output

B. Understands and applies knowledge of procedures, event-driven programs, usability, data structures, debugging, documenting and reviewing code, libraries and APIs, IDEs, and programming language paradigms, including object-oriented concepts
1. Understand how to write and call procedures with parameters and return values
   a. trace code and indicate the output printed or the value of variables after code segment execution
   b. indicate inputs that produce given outputs for a code segment
   c. describe what a program does or choose the code segment that correctly implements a stated intended purpose
   d. identify missing code in a code segment with a stated intended purpose
   e. identify equivalent statements or code segments
   f. place statements in appropriate order to create a correct program
   g. trace code when references to objects and arrays are passed to procedures
   h. trace code that includes nested procedure calls
2. Know the concepts of event-driven programs that respond to external events (e.g., sensors, messages, clicks)
   a. trace code and indicate the output printed or the value of variables after code segment execution
   b. indicate inputs that produce given outputs for a code segment
   c. describe what a program does or choose the code segment that correctly implements a stated intended purpose
   d. identify missing code in a code segment with a stated intended purpose
   e. identify possible errors due to asynchronous events
   f. identify aspects of concurrency in event-driven programming
3. Be familiar with usability and user experience (e.g., ease of use and accessibility)
   a. identify code that improves on given code in terms of usability or user experience
   b. identify meaningful error messages
   c. identify features that improve accessibility
4. Be familiar with dictionaries/maps, stacks, and queues
   a. identify a data structure based on a description of behavior or appropriate use
   b. given goals, constraints, or context, identify the most appropriate data structure
   c. trace code that uses a particular data structure
5. Understand how to use debugging techniques and appropriate test cases
   a. identify which test cases are most useful for given code
   b. differentiate between different types of errors (e.g., overflow, round-off, syntax, runtime, compile-time, logic)
   c. describe useful debugging techniques (e.g., where to put print statements)
   d. differentiate between empirical testing and proof
   e. identify errors in code and solutions to those errors

6. Be familiar with characteristics of well-documented computer programs that are usable, readable, and modular
   a. identify characteristics of good documentation
   b. identify good and poor documentation practices in context

7. Be familiar with techniques to obtain and use feedback to produce high-quality code (e.g., code reviews, peer feedback, end user feedback)
   a. identify situations in which each of the three listed techniques are useful

8. Know how to use libraries and APIs
   a. identify correct call(s) and use of return values given an API definition
   b. identify reasons to use or not use libraries in place of writing original code
   c. identify applications (e.g., math libraries, random number generation) that use APIs

9. Understand programming techniques to validate correct input and detect incorrect input
   a. identify effective input data validation strategies
   b. compare data validation (proper range and format) and data verification (e.g., password verification)
   c. identify improvements to code for which data validation is required

10. Be familiar with the features and capabilities of integrated development environments (IDEs)
    a. identify components of IDEs
    b. identify benefits and drawbacks of using IDEs
    c. identify the costs and benefits of context editors

11. Be familiar with the differences between low- and high-level programming languages
    a. identify characteristics of low- and high-level languages

12. Be familiar with different programming paradigms
    a. identify the terminology of procedural programming
    b. identify the terminology of object-oriented programming
    c. compare programming paradigms

13. Know object-oriented programming concepts
    a. identify classes, instance variables, and methods given a diagram
    b. identify the benefits of inheritance and encapsulation
    c. identify distinctions between overloading and overriding

14. Be familiar with program compilation and program interpretation
    a. identify differences between compilation and interpretation
    b. identify differences between source code and object code

IV. Data

A. Understands and applies knowledge of digitalization, data encryption and decryption, and computational tools

1. Understand bits as the universal medium for expressing digital information
   a. perform calculations, using bits and bytes
   b. determine the number of bits and bytes required to store a given amount of data
   c. given the description of an encoding scheme, encode or decode data
   d. describe lossy and lossless data compression
   e. explain why binary numbers are fundamental to the operation of computer systems

2. Be familiar with concepts of data encryption and decryption
   a. distinguish between encoding and encryption
   b. identify trade-offs in the use of data encryption
3. Know how to use computational tools, including spreadsheets, to analyze data in order to discover, explain, and visualize patterns, connections, and trends
   a. transform data to make it more useful
   b. identify specific data or characteristics of specific data that need to be removed or modified before an entire data set can be used
   c. describe the use of spreadsheet operations (e.g., formulas, filters, sorts, charts, graphs) to analyze and visualize data

B. Understands and applies knowledge of simulation, modeling, and manipulation of data
1. Be familiar with the use of computing in simulation and modeling
   a. describe questions that can be answered with a given simulation, or explain what data and process are required in a simulation in order to answer a given question
   b. trace code in a simulation context
   c. identify missing code in a simulation context
   d. identify the impact of changes to simulations (e.g., more or fewer variables, more or less data)
   e. identify applications of simulation and modeling
2. Be familiar with methods to store, manage, and manipulate data
   a. use terminology and concepts of files and databases
   b. identify measures of file size (e.g., byte, kilo, mega, giga, tera, peta)
   c. identify issues connected with the storage requirements of computing applications, including scale, redundancy, and backup
3. Be familiar with a variety of computational methods for data collection, aggregation, and generation
   a. identify the benefits of working with publicly available data sets
   b. identify the types of data generated by surveys and sensors
   c. identify examples of crowdsourcing and citizen science
   d. identify appropriate data-collection methods for a given context and purpose

V. Computing Systems and Networks
A. Understands and applies knowledge of operating systems, computing systems, communication between devices, and cloud computing
1. Know that operating systems are programs that control and coordinate interactions between hardware and software components
   a. identify hardware components and their functions
   b. identify software components and their functions
   c. identify common operating systems tasks
   d. identify resource issues that have an impact on functionality
2. Be familiar with computing systems embedded in everyday objects (e.g., Internet of Things [IoT], ATMs, medical devices)
   a. describe what an embedded system is
   b. define what the IoT is and how it is used
   c. describe how sensors are used in embedded systems
3. Know the capabilities, features, and uses of different types of computing systems (e.g., desktop, mobile, cluster)
   a. identify capabilities, features, and uses for each type of computer system
   b. identify criteria to evaluate and compare computing systems
4. Be familiar with computers as layers of abstraction from hardware (e.g., logic gates, chips) to software (e.g., system software, applications)
   a. identify appropriate abstraction layers for hardware and software components
5. Be familiar with the steps required to execute a computer program (fetch-decode-execute cycles)
   a. describe what happens during fetch, decode, and execute, including the order of the steps in the cycle
6. Be familiar with trade-offs between local, network, and cloud computing and storage
   a. identify advantages and disadvantages in terms of performance, cost, security, reliability, and collaboration
   b. identify means of storing binary data
7. Be familiar with communication between devices
   a. identify and compare wireless communication systems
   b. identify and compare wired communication systems
   c. identify and compare network types

B. Understands and applies knowledge of networks, including security issues and the Web

1. Know components of networks
   a. identify network hardware devices and their functions
   b. describe possible abstraction models of networks
2. Be familiar with factors that have an impact on network functionality
   a. define basic terminology (e.g., bandwidth, load, latency)
   b. estimate necessary bandwidth and data size for a given situation
   c. identify critical resources for a given situation
3. Be familiar with how Internet and Web protocols work
   a. describe the purpose of protocols and identify common Internet and Web protocols
   b. compare IPv4 and IPv6
   c. identify and describe the basic parts of a URL (e.g., protocol, subdomain, domain name, port, path)
   d. describe the hierarchical structure of names in the domain name system (DNS)
   e. describe the purpose and function of IP addressing
   f. identify how Internet protocols address reliability, redundancy, and error handling
4. Be familiar with digital and physical strategies for maintaining security
   a. identify characteristics of strong passwords (e.g., length, bits per character)
   b. identify digital and physical security strategies
   c. identify trade-offs in the use of security measures (e.g., encryption, decryption, digital signatures and certificates)
5. Be familiar with concepts of cybersecurity
   a. identify and define the five pillars of cybersecurity: confidentiality, integrity, availability, nonrepudiation, and authentication
6. Be familiar with the components that make up the Web (e.g., HTTP, HTML, browsers, servers, clients)
   a. identify the uses of markup languages
   b. identify the purposes of browsers, servers, and clients
**Pseudocode Notation**

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

<table>
<thead>
<tr>
<th>Explanation</th>
<th>Notation</th>
</tr>
</thead>
<tbody>
<tr>
<td>Assignment operator</td>
<td>←</td>
</tr>
<tr>
<td>Arithmetic operators</td>
<td>+ - / * ^ %</td>
</tr>
<tr>
<td>Note that / indicates floating point division unless stated otherwise.</td>
<td></td>
</tr>
<tr>
<td>Relational operators</td>
<td>== &lt; &gt; ≤ ≥ ≠</td>
</tr>
<tr>
<td>Logical operators</td>
<td>and or not</td>
</tr>
<tr>
<td>String concatenation operator</td>
<td>+</td>
</tr>
<tr>
<td>Boolean values</td>
<td>true false</td>
</tr>
<tr>
<td>Null</td>
<td>null</td>
</tr>
<tr>
<td>Comments</td>
<td>// this is a single-line comment</td>
</tr>
<tr>
<td>Placeholder for missing code</td>
<td>For example, /* missing code <em>/ /</em> missing condition */</td>
</tr>
<tr>
<td>Print</td>
<td>print arg</td>
</tr>
<tr>
<td>A comment is used where necessary to indicate if a line feed or blank is appended to the argument.</td>
<td></td>
</tr>
<tr>
<td>Data types</td>
<td>boolean char double float int int[] int[][] short String</td>
</tr>
<tr>
<td>Array initialization and reference</td>
<td>int[] a ← (1, 2, 3) int b[0..2] ← (1, 2, 3) int[][] c a[0]</td>
</tr>
</tbody>
</table>
### Explanation

Conditional statements: Indentation and `end if` statements are significant.

Example:
```
if ( x > 10 )
    print "big number"
else
    print "small number"
end if
```

Iterative statements: Indentation and `end` statements are significant

```
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 )
```

Procedures: Indentation and `end` statements are significant.

The return type is indicated in the procedure header and is based on the value returned by the procedure or is `void` if the procedure does not return a value.

```
int procedureName ( arg1, arg2, ... )
    block of statements
return value
end procedureName

void procedureName ( arg1, arg2, ... )
    block of statements
end procedureName
```

### Notation

- `if ( condition )` followed by a block of statements `end if`
- `if ( condition )` followed by a block of statements `else` followed by another block of statements `end if`
- `for ( initialization; condition; increment )` followed by a block of statements `end for`
- `while ( condition )` followed by a block of statements `end while`
- `do` followed by a block of statements `while ( condition )`
- `repeat` followed by a block of statements `until ( condition )`
- `int procedureName ( arg1, arg2, ... )` followed by a block of statements `return value` `end procedureName`
- `void procedureName ( arg1, arg2, ... )` followed by a block of statements `end procedureName`

### Classes

- `class` followed by a class name
- `variable declarations` followed by `procedures`
- `end class` followed by a class name

### Object-oriented keywords

- `extends`
- `new`
- `public`
- `private`
Step 2: Familiarize Yourself with Test Questions

2. Familiarize Yourself with Test Questions

Become comfortable with the types of questions you’ll find on the Praxis tests

The Praxis assessments include a variety of question types: constructed response (for which you write a response of your own); selected response, for which you select one or more answers from a list of choices or make another kind of selection (e.g., by clicking on a sentence in a text or by clicking on part of a graphic); and numeric entry, for which you enter a numeric value in an answer field. You may be familiar with these question formats from taking other standardized tests. If not, familiarize yourself with them so you don’t spend time during the test figuring out how to answer them.

Understanding Computer-Delivered Questions

Questions on computer-delivered tests are interactive in the sense that you answer by selecting an option or entering text on the screen. 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.

For most questions, you respond by clicking an oval to select a single answer from a list of answer choices.

However, interactive question types may also ask you to respond by:

- Clicking more than one oval to select answers from a list of choices.
- Typing in an entry box. When the answer is a number, you may be asked to enter a numerical answer. 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 select your answers by clicking on a location (or locations) on a graphic such as a map or chart, as opposed to choosing your answer from a list.
- Clicking on sentences. In questions with reading passages, you may be asked to choose your answers 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 select answers from a list of choices and drag your answers to the appropriate location in a table, paragraph of text or graphic.
- Selecting answer choices from a drop-down menu. You may be asked to choose answers by selecting choices from a drop-down menu (e.g., to complete a sentence).

Remember that with every question you will get clear instructions.

Perhaps the best way to understand computer-delivered questions is to view the Computer-delivered Testing Demonstration on the Praxis web site to learn how a computer-delivered test works and see examples of some types of questions you may encounter.
Understanding Selected-Response Questions

Many selected-response questions begin with the phrase “which of the following.” Take a look at this example:

Which of the following is a flavor made from beans?

(A) Strawberry
(B) Cherry
(C) Vanilla
(D) Mint

How would you answer this question?

All of the answer choices are flavors. Your job is to decide which of the flavors is the one made from beans.

Try following these steps to select the correct answer.

1) Limit your answer to the choices given. You may know that chocolate and coffee are also flavors made from beans, but they are not listed. Rather than thinking of other possible answers, focus only on the choices given (“which of the following”).

2) Eliminate incorrect answers. You may know that strawberry and cherry flavors are made from fruit and that mint flavor is made from a plant. That leaves vanilla as the only possible answer.

3) Verify your answer. You can substitute “vanilla” for the phrase “which of the following” and turn the question into this statement: “Vanilla is a flavor made from beans.” This will help you be sure that your answer is correct. If you’re still uncertain, try substituting the other choices to see if they make sense. You may want to use this technique as you answer selected-response questions on the practice tests.

Try a more challenging example

The vanilla bean question is pretty straightforward, but you’ll find that more challenging questions have a similar structure. For example:

Entries in outlines are generally arranged according to which of the following relationships of ideas?

(A) Literal and inferential
(B) Concrete and abstract
(C) Linear and recursive
(D) Main and subordinate

You’ll notice that this example also contains the phrase “which of the following.” This phrase helps you determine that your answer will be a “relationship of ideas” from the choices provided. You are supposed to find the choice that describes how entries, or ideas, in outlines are related.

Sometimes it helps to put the question in your own words. Here, you could paraphrase the question in this way: “How are outlines usually organized?” Since the ideas in outlines usually appear as main ideas and subordinate ideas, the answer is (D).
QUICK TIP: Don’t be intimidated by words you may not understand. It might be easy to be thrown by words like “recursive” or “inferential.” Read carefully to understand the question and look for an answer that fits. An outline is something you are probably familiar with and expect to teach to your students. So slow down, and use what you know.

Watch out for selected-response questions containing “NOT,” “LEAST,” and “EXCEPT”

This type of question asks you to select the choice that does not fit. You must be very careful because it is easy to forget that you are selecting the negative. This question type is used in situations in which there are several good solutions or ways to approach something, but also a clearly wrong way.

How to approach questions about graphs, tables, or reading passages

When answering questions about graphs, tables, or reading passages, provide only the information that the questions ask for. In the case of a map or graph, you might want to read the questions first, and then look at the map or graph. In the case of a long reading passage, you might want to go ahead and read the passage first, noting places you think are important, and then answer the questions. Again, the important thing is to be sure you answer the questions as they refer to the material presented. So read the questions carefully.

How to approach unfamiliar formats

New question formats are developed from time to time to find new ways of assessing knowledge. Tests may include audio and video components, such as a movie clip or animation, instead of a map or reading passage. Other tests may allow you to zoom in on details in a graphic or picture.

Tests may also include interactive questions. These questions take advantage of technology to assess knowledge and skills in ways that standard selected-response questions cannot. 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.

QUICK TIP: Don’t make the questions more difficult than they are. Don’t read for hidden meanings or tricks. There are no trick questions on Praxis tests. They are intended to be serious, straightforward tests of your knowledge.

Understanding Constructed-Response Questions

Constructed-response questions require you to demonstrate your knowledge in a subject area by creating your own response to particular topics. Essays and short-answer questions are types of constructed-response questions.

For example, an essay question might present you with a topic and ask you to discuss the extent to which you agree or disagree with the opinion stated. You must support your position with specific reasons and examples from your own experience, observations, or reading.

Take a look at a few sample essay topics:

- “Celebrities have a tremendous influence on the young, and for that reason, they have a responsibility to act as role models.”
- “We are constantly bombarded by advertisements—on television and radio, in newspapers and magazines, on highway signs, and the sides of buses. They have become too pervasive. It’s time to put limits on advertising.”
- “Advances in computer technology have made the classroom unnecessary, since students and teachers are able to communicate with one another from computer terminals at home or at work.”
Keep these things in mind when you respond to a constructed-response question

1) **Answer the question accurately.** Analyze what each part of the question is asking you to do. If the question asks you to describe or discuss, you should provide more than just a list.

2) **Answer the question completely.** If a question asks you to do three distinct things in your response, you should cover all three things for the best score. Otherwise, no matter how well you write, you will not be awarded full credit.

3) **Answer the question that is asked.** Do not change the question or challenge the basis of the question. You will receive no credit or a low score if you answer another question or if you state, for example, that there is no possible answer.

4) **Give a thorough and detailed response.** You must demonstrate that you have a thorough understanding of the subject matter. However, your response should be straightforward and not filled with unnecessary information.

5) **Reread your response.** Check that you have written what you thought you wrote. Be sure not to leave sentences unfinished or omit clarifying information.

**QUICK TIP:** You may find that it helps to take notes on scratch paper so that you don’t miss any details. Then you’ll be sure to have all the information you need to answer the question.
3. Practice with Sample Test Questions

Answer practice questions and find explanations for correct answers

Computer Delivery

This test is available on computer. The following sample question provides a preview of the actual screen used in a computer-delivered test. For the purposes of this Study Companion, the sample questions in this chapter are shown as they would appear in a paper-delivered test.

What quantity of oxygen, $O_2$, contains very nearly the same number of molecules as 50.0 grams of water, $H_2O$?

- 64.0 grams
- 32.0 grams
- 16.0 grams
- 8.0 grams
Sample Test Questions

The sample questions that follow illustrate the kinds of questions on the test. They are not, however, representative of the entire scope of the test in either content or difficulty. Answers with explanations follow the questions.

Directions: Each of the questions or statements below is followed by suggested answers or completions. Select the answer or answers that are best in each case.

1. For the following question, select all the answer choices that apply.

A technology company has developed an application that allows users to combine a text file and a recording of a person speaking to create an accurate audio representation of the person speaking the text. Which of the following are possible uses of the application? Select all that apply.

(A) Audio versions of unrecorded speeches delivered by famous individuals could be produced from transcripts and recorded speeches delivered by the same individual.

(B) An audio file could be fabricated in which an individual seems to be uttering slanderous statements about another individual.

(C) Security systems based on voice activation could be circumvented by unauthorized users.

2. A programmer uses code published online under a Creative Commons Attribution (CC BY) license in a commercial product. Which of the following best describes an acceptable use of the code?

(A) Copying code from the online source into the programmer’s product without any other actions

(B) Copying code from the online source into the programmer’s product and limiting the copied code to ten code lines

(C) Copying code from the online source into the programmer’s product and changing all variable names

(D) Copying code from the online source into the programmer’s product and crediting the original author in the manner indicated by the license

3. Which of the following best describes the primary way in which a distributed denial-of-service (DDoS) attack differs from a denial-of-service (DoS) attack?

(A) The goal of the attack

(B) The number of computers being attacked

(C) The number of computers launching the attack

(D) The time period in which the attack occurs

4. Consider the following list.

• Assembly language
• Block-based programming language
• Logic gate
• Machine language

Which of the following arranges the list in order from highest level of abstraction to lowest level of abstraction?

(A) Block-based programming language, assembly language, machine language, logic gate

(B) Block-based programming language, machine language, assembly language, logic gate

(C) Block-based programming language, machine language, logic gate, assembly language

(D) Machine language, block-based programming language, assembly language, logic gate
5. Chloe is playing a game in which she rolls a pair of dice 6 times. Her initial score is 0, and after each roll her score is recalculated. The table shows the outcomes of each roll and Chloe’s recalculated score.

<table>
<thead>
<tr>
<th>Roll</th>
<th>1st</th>
<th>2nd</th>
<th>3rd</th>
<th>4th</th>
<th>5th</th>
<th>6th</th>
</tr>
</thead>
<tbody>
<tr>
<td>Die One</td>
<td>3</td>
<td>1</td>
<td>6</td>
<td>1</td>
<td>6</td>
<td>2</td>
</tr>
<tr>
<td>Die Two</td>
<td>4</td>
<td>5</td>
<td>2</td>
<td>6</td>
<td>6</td>
<td>2</td>
</tr>
<tr>
<td>Score</td>
<td>7</td>
<td>13</td>
<td>13</td>
<td>13</td>
<td>0</td>
<td>4</td>
</tr>
</tbody>
</table>

Consider the following pseudocode segment.

```plaintext
int score ← 0
int dieOne ← 0
int dieTwo ← 0
for ( int i ← 0; i < 6; i ← i + 1 )
    dieOne ← get DieOneValue ()  // returns the value of the first die
    dieTwo ← get DieTwoValue ()  // returns the value of the second die
    score ← newScore ( dieOne, dieTwo, score )
end for
```

Based on the data in the table, which of the following correctly implements the `newScore` procedure?

(A) ```plaintext
int newScore ( int diceOne, int diceTwo, int oldScore )
return oldScore + diceOne + diceTwo
end newScore
```

(B) ```plaintext
int newScore ( int diceOne, int diceTwo, int oldScore )
oldScore ← oldScore + diceOne + diceTwo
if ( ( diceOne == 6 ) or ( diceTwo == 6 ) )
    oldScore ← oldScore - diceOne - diceTwo
else
    if ( ( diceOne == 6 ) and ( diceTwo == 6 ) )
        oldScore ← 0
    end if
end if
return oldScore
end newScore
```

(C) ```plaintext
int newScore ( int diceOne, int diceTwo, int oldScore )
if ( ( diceOne ≠ 6 ) and ( diceTwo ≠ 6 ) )
    return oldScore + diceOne + diceTwo
else
    if ( ( diceOne == 6 ) and ( diceTwo == 6 ) )
        return 0
    else
        return oldScore
    end if
end if
end newScore
```

(D) ```plaintext
int newScore ( int diceOne, int diceTwo, int oldScore )
if ( diceOne == diceTwo )
    return 0
else
    if ( ( diceOne == 6 ) or ( diceTwo == 6 ) )
        return oldScore
    else
        return oldScore + diceOne + diceTwo
    end if
end if
end newScore
```
6. Which of the following is the hexadecimal representation of the decimal number \(231_{10}\)?

(A) \(17_{16}\)

(B) \(E4_{16}\)

(C) \(E7_{16}\)

(D) \(F4_{16}\)

7. A sales representative has a list of clients to visit during an upcoming sales trip. No two clients live in the same city, and each client will be visited one time. The sales representative will return to the starting city at the end of the trip. To minimize travel expenses, a programmer at the sales representative's company has implemented an algorithm that generates all possible orderings of the clients' cities and then evaluates each ordering with respect to travel expenses. Which of the following best describes the running time of the algorithm in terms of the number of cities?

(A) Factorial

(B) Linear

(C) Logarithmic

(D) Quadratic

8. Consider the following pseudocode procedure, which sorts an integer array \(a\[\]\) of length \(l\). The first element of \(a\[\]\) is at index 0. A call \(\text{swap} (a[\ ], i, j)\) swaps the values of \(a[i]\) and \(a[j]\).

```plaintext
void sort ( int[] arr, int len )
    int pos ← 0
    while ( pos < len )
        if ( pos == 0 )
            pos ← pos + 1
        else
            if ( arr[pos] > arr[pos - 1] )
                pos ← pos + 1
            else
                swap ( arr, pos, pos - 1 )
                pos ← pos - 1
            end if
        end if
    end while
end sort
```

If \(a\) originally contains the values \(\{2, 1, 5, 3, 4\}\), what will the values in \(a\) be after 6 iterations of the while loop?

(A) \(\{1, 2, 3, 4, 5\}\)

(B) \(\{1, 2, 3, 5, 4\}\)

(C) \(\{1, 2, 5, 3, 4\}\)

(D) \(\{2, 1, 5, 3, 4\}\)
9. Consider the recursive pseudocode procedure $f$, which is intended to return the product of consecutive integers from 5 to $n$, inclusive, for $n \geq 5$ and to return zero otherwise. For example, $f(7)$ returns 210, which is $5 \times 6 \times 7$.

```java
int f (int n)
    if (n < 5)
        return 0
    else
        if (n == 5)
            return 5
        else
            /* missing statement */
        end if
    end if
end f
```

Which of the following could replace /* missing statement */ so that the procedure $f$ works as intended?

(A) return $n \times f(n - 1)$
(B) return $n \times f(n - 5)$
(C) return $(n - 5) \times f(n)$
(D) return $(n - 5) + f(n - 1)$

10. Consider the pseudocode procedure $\text{findMax}$, which is intended to return the largest value in an integer array $\text{numList}$ of length $n$. The first element of $\text{numList}$ is at index 0.

```java
int findMax (int[] numList, int n)
    int max ← numList[0]
    int i ← 1
    while (i < n)
        if (/* missing condition */)
            max ← numList[i]
        end if
        i ← i + 1
    end while
    return max
end findMax
```

Which of the following could replace /* missing condition */ so that $\text{findMax}$ works as intended?

(A) $\text{numList}[i] < n$
(B) $\text{numList}[i] < \text{max}$
(C) $\text{numList}[i] > \text{max}$
(D) $\text{numList}[i] > \text{numList}[n - 1]$
11. Consider the following pseudocode procedure, which is intended to print the odd integers from 1 up to \( n \).

```plaintext
// precondition: n is a positive integer
void f ( int n )
    for ( int c ← 1; c ≤ n; c ← c + 1 )
        if ( ( c / 2 ) ≠ 0 )
            print c
        end if
    end for
end f
```

Which of the following is a true statement?

(A) The procedure works correctly and prints all the odd integers from 1 up to \( n \).

(B) The procedure does not work correctly; a new variable needs to be declared inside the loop to test for odd numbers.

(C) The procedure does not work correctly; the variable \( c \) needs to be declared before the for loop.

(D) The procedure does not work correctly; there is a logic error in the if condition, which should say \( c \% 2 \) instead of \( c / 2 \).

12. Consider the following pseudocode procedure.

```plaintext
void mystery ( int n )
    while ( n ≠ 1 )
        if ( ( n % 2 ) == 1 )
            n ← 3 * n + 1
        else
            n ← n / 2
        end if
        print ( n ) // print a space after the number
    end while
end mystery
```

What is printed by the call `mystery ( 6 )`?

(A) 3 10 5 16 8 4 2

(B) 3 10 5 16 8 4 2 1

(C) 10 5 16 8 4 2 1

(D) Infinitely many numbers are printed because the while loop does not terminate.

13. Consider the following pseudocode Boolean expression, where `age` and `height` are two properly declared and initialized integer variables.

\( ( \text{age} > 10 ) \land ( \text{height} > 36 ) \)

Which of the following is an equivalent Boolean expression?

(A) \( \neg ( ( \text{age} < 10 ) \land ( \text{height} < 36 ) ) \)

(B) \( \neg ( ( \text{age} ≤ 10 ) \land ( \text{height} ≤ 36 ) ) \)

(C) \( \neg ( ( \text{age} < 10 ) \lor ( \text{height} < 36 ) ) \)

(D) \( \neg ( ( \text{age} ≤ 10 ) \lor ( \text{height} ≤ 36 ) ) \)
14. Consider a class **String** with the methods shown.

<table>
<thead>
<tr>
<th>Procedure</th>
<th>Explanation</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>String toLowerCase ()</strong></td>
<td>Converts the characters in the string to lowercase and returns the string.</td>
</tr>
<tr>
<td><strong>String toUpperCase ()</strong></td>
<td>Converts the characters in the string to uppercase and returns the string.</td>
</tr>
<tr>
<td><strong>String substring ( int begin, int end )</strong></td>
<td>Returns the substring of the string from position \begin{equation} \text{begin} \end{equation} to \begin{equation} \text{end} - 1 \end{equation} or returns &quot;&quot; if \begin{equation} \text{begin} \end{equation} is less than zero or \begin{equation} \text{begin} \geq \text{end} \end{equation} or \begin{equation} \text{end} \end{equation} is greater than the length of the string. The first character in the string is at position 0.</td>
</tr>
<tr>
<td><strong>int length ()</strong></td>
<td>Returns the length of the string.</td>
</tr>
</tbody>
</table>

Which of the following pseudocode segments would produce the output " aPPLESAUCE "?

(A) **String** value ← " Applesauce "
    value ← value.substring ( 0, 1 ).toLowerCase () +
    value.substring ( 1, value.length () ).toUpperCase ()
    print value

(B) **String** value ← " Applesauce "
    value ← value.substring ( 0, 2 ).toLowerCase () +
    value.substring ( 1, value.length () ).toUpperCase ()
    print value

(C) **String** value ← " Applesauce "
    value ← value.substring ( 0, 1 ).toUpperCase () +
    value.substring ( 1, value.length () ).toLowerCase ()
    print value

(D) **String** value ← " Applesauce "
    value ← value.toUpperCase ()
    value ← value.toLowerCase ()
    print value
15. Consider the following pseudocode procedure.

```plaintext
int mystery ( int n )
    int temp ← 0
    for ( int c ← 1; c ≤ n; c ← c + 1 )
        if ( ( c % 3 ) == 0 )
            temp ← temp + c
    end if
end for
return temp
end mystery
```

Which of the following best describes procedure `mystery`?

(A) It returns a list of numbers from 1 to n.
(B) It prints every third number from 1 to n.
(C) It returns the sum of the numbers from 1 to n.
(D) It returns the sum of the multiples of 3 from 1 to n.

16. Consider the following three scenarios, which could be modeled using three data structures—dictionary/map, queue, and stack.

**Scenario 1**: Cars line up in a single lane at a car wash. As each driver reaches the entrance of the car wash, the driver gets out of the car. An automatic pulley moves the car through the car wash as the driver walks to the exit of the car wash to pay and pick up the car.

**Scenario 2**: Contestants auditioning for a reality television talent show are assigned a unique numeric ID upon completing a registration form.

**Scenario 3**: Tennis balls are sold in a cylindrical can that can hold a maximum of 3 balls, where each ball except the bottom one rests on top of the ball below it. The 3 balls are placed in the can one at a time through one opening at the top of the can. The 3 balls are removed from the can one at a time through the same opening.

Which of the following shows the data structures that best model the scenarios?

<table>
<thead>
<tr>
<th>Scenario 1</th>
<th>Scenario 2</th>
<th>Scenario 3</th>
</tr>
</thead>
<tbody>
<tr>
<td>(A) Dictionary/map</td>
<td>Queue</td>
<td>Stack</td>
</tr>
<tr>
<td>(B) Dictionary/map</td>
<td>Stack</td>
<td>Queue</td>
</tr>
<tr>
<td>(C) Queue</td>
<td>Dictionary/map</td>
<td>Stack</td>
</tr>
<tr>
<td>(D) Stack</td>
<td>Queue</td>
<td>Dictionary/map</td>
</tr>
</tbody>
</table>
17. Consider the following pseudocode segment with integer variables, which is intended to determine the largest value among variables \(a\), \(b\), and \(c\), and to store that value in variable \(max\). The code segment does not work as intended.

\[
\begin{align*}
\text{max} & \leftarrow -1 \\
\text{if} & \ (a > b) \\
& \quad \text{if} \ (a > c) \\
& \quad \text{max} \leftarrow a \\
& \quad \text{end if} \\
\text{else} & \\
& \quad \text{if} \ (b > c) \\
& \quad \text{max} \leftarrow b \\
& \quad \text{else} \\
& \quad \text{max} \leftarrow c \\
& \quad \text{end if} \\
\text{end if}
\end{align*}
\]

Which of the following test cases can be used to demonstrate that the code segment does not work as intended?

(A) \(a \leftarrow 20\)  
\(b \leftarrow 15\)  
\(c \leftarrow 15\)  

(B) \(a \leftarrow 20\)  
\(b \leftarrow 15\)  
\(c \leftarrow 25\)  

(C) \(a \leftarrow 20\)  
\(b \leftarrow 25\)  
\(c \leftarrow 25\)  

(D) \(a \leftarrow 20\)  
\(b \leftarrow 25\)  
\(c \leftarrow 30\)  

18. Huffman coding assigns unique variable-length codes to input values based on the frequency of occurrence of each value. Frequently occurring values are assigned codes that contain fewer bits than values that occur less frequently, which are assigned codes that contain more bits. Which of the following best describes an appropriate use of Huffman coding?

(A) Decryption  
(B) Efficient sorting  
(C) Lossless compression  
(D) Lossy compression  

19. Which of the following spreadsheet functions would be most useful for detecting improbably high or low values that have become part of a data set as a result of data entry errors?

(A) A function that averages numeric values in a column or row  
(B) A function that counts the values in a column or row  
(C) A function that rounds a numeric value  
(D) A function that sorts values in a column or row
20. A computer simulation is created to simulate the growth of a certain plant species in different conditions. Which of the following actions could be used to validate the model used in the simulation?

(A) Express the simulation software using both recursive and iterative algorithms. Compare the results of the recursive algorithm to those of the iterative algorithm.

(B) Perform real-world experiments on the plant species’ growth in different environments. Compare the experimental results to the results provided by the simulation.

(C) Remove any unnecessary details from the model. Compare the running times of the original simulation and the simplified simulation.

(D) Run the simulation software on multiple devices. Compare the results obtained from each of the devices.

21. Consider the following assumptions about electronic storage for the text in all the books of a university's libraries.

- The libraries at the university collectively contain 3 million books.
- A book contains an average of 400 pages.
- A page contains an average of 50 lines.
- A line on a page contains an average of 10 words.
- A word contains an average of 5 letters/characters.
- A letter/character is represented by 1 byte.

Based on the given assumptions, which of the following is the unit in which the electronic storage required for the text in all the books of the university's libraries would best be measured?

(A) Megabyte (2^20 or approximately 10^6 bytes)
(B) Gigabyte (2^30 or approximately 10^9 bytes)
(C) Terabyte (2^40 or approximately 10^12 bytes)
(D) Petabyte (2^50 or approximately 10^15 bytes)

22. Which of the following is an example of the use of a device on the Internet of Things (IoT)?

(A) A car alerts a driver that it is about to hit an object.
(B) A hiker uses a GPS watch to keep track of her position.
(C) A refrigerator orders milk from an online delivery service when the milk in the refrigerator is almost gone.
(D) A runner uses a watch with optical sensors to monitor his heart rate.

23. What is the role of the compiler in the process of developing executable software?

(A) Managing specification files created as part of the development process
(B) Running and testing the executable created by the programmer
(C) Tracking older versions of the software in case an error is found and the software needs to be reverted to an earlier form
(D) Translating a program written in an abstract, high-level language into a program with the same behavior expressed in machine code
24. Which of the following best describes a Web server?
   (A) A computer system that delivers Web pages to clients
   (B) A computer system that determines the shortest path between two computers over the Internet
   (C) A computer system running software that provides a user-friendly interface for creating Web pages
   (D) A computer system that translates domain names to IP addresses

25. Which pillar of cybersecurity is compromised when someone logs into a system using a stolen login and password?
   (A) Authentication
   (B) Confidentiality
   (C) Integrity
   (D) Nonrepudiation
Answers to Sample Questions

1. The correct answers are (A), (B), and (C). All three choices are possible uses of the application. Choice (A): Using the application to combine a transcript of an unrecorded speech with existing recordings of an individual would create an audio reconstruction of the unrecorded speech. Choice (B): Using the application to combine false statements with existing recordings of an individual would create an audio file in which the individual seems to be uttering the false statements. Choice (C): The application could be used to create voice commands that could fool voice activation systems.

2. The correct answer is (D). The Creative Commons Attribution (CCBY) license allows anyone to use, revise, and distribute the code, even commercially, as long as the original author is credited properly.

3. The correct answer is (C). A DDoS attack is a DoS attack that originates from many different sources.

4. The correct answer is (A). Of the entries in the list, block-based programming language has the highest level of abstraction, logic gate has the lowest level of abstraction, and the other two entries are in between. Machine language is closer to hardware than is assembly language, so assembly language has a higher level of abstraction than does machine language.

5. The correct answer is (C). An analysis of the information in the table shows that: (i) when neither of the two dice is a 6, the score increases by the sum of the values of the two dice; (ii) when both of the dice are a 6, the score becomes 0; and (iii) when one but not both of the two dice is a 6, the score remains the same. Only the code segment in choice (C) is compatible with these observations—the first return statement in option (C) corresponds to case (i) above; the second return statement corresponds to case (ii); and the third return statement corresponds to case (iii).

6. The correct answer is (C). Since $231 = 14 \times 16 + 7$, the decimal number $231_{10}$ is equivalent to the hexadecimal number $E7_{16}$.

7. The correct answer is (A). The algorithm solves the traveling salesman problem using a brute-force approach, which enumerates all possible trips and identifies the least expensive one. The number of possible orderings of the clients’ cities is factorial in the number of cities, so the running time of the algorithm is factorial.

8. The correct answer is (B). At the end of the first iteration, the value of pos is 1 and the array is unchanged. At the end of the second iteration, the value of pos is 0 and the array is $(1, 2, 5, 3, 4)$. At the end of the third iteration, the value of pos is 1 and the array is $(1, 2, 5, 3, 4)$. At the end of the fourth iteration, the value of pos is 2 and the array is $(1, 2, 5, 3, 4)$. At the end of the fifth iteration, the value of pos is 3 and the array is $(1, 2, 5, 3, 4)$. At the end of the sixth iteration, the value of pos is 2 and the array is $(1, 2, 3, 5, 4)$.

9. The correct answer is (A). When the execution of the code reaches the missing statement, the value of $n$ is greater than or equal to 6. When $n \geq 6$, the value of $f(n)$ is the product of consecutive integers from 5 to $n$, which is the product of the consecutive integers from $n - 1$ down to 5, which is $n$ times the product of the consecutive integers from $n - 1$ down to 5, which is $n$ times $f(n - 1)$. Therefore, when $n \geq 6$, we have the recurrence relation $f(n) = n \cdot f(n - 1)$.

10. The correct answer is (C). At the beginning of each iteration of the while loop, the value of max is the largest value in the subarray numList[0..i-1]. The missing statement is numList[i] > max, which is equivalent to saying that numList[i] is greater than all the elements in the subarray numList[0..i-1]. If the comparison is true, the value of numList[i] is assigned to max; if not, the value of max is unchanged.

11. The correct answer is (D). The procedure does not work correctly—as is, the procedure does not print anything when $n$ is 1 and prints all consecutive integers from 2 to $n$ when $n \geq 2$. To fix the procedure, the integer division operator / in the if condition needs to be replaced with the modulus (remainder) operator %.
12. The correct answer is (B). The if-else statement checks whether the value of \( n \) is an odd integer; if \( n \) is odd, it replaces it with one more than its triple; if \( n \) is even, it halves it. Since the initial value of \( n \) is 6, an even number, the procedure prints 10. Since 10 is an even number, the procedure prints 5. Since 5 is an odd number, the procedure prints 10. Since 10 is an even number, the procedure prints 5. Since 5 is an odd number, the procedure prints 10. Since 10 is an even number, the procedure prints 5. Since 5 is an odd number, the procedure prints 10. Since 10 is an even number, the procedure prints 5. Since 5 is an odd number, the procedure prints 10. Since 10 is an even number, the procedure prints 5. Since 5 is an odd number, the procedure prints 10. Since 10 is an even number, the procedure prints 5. Since 5 is an odd number, the procedure prints 10. Since 10 is an even number, the procedure prints 5. Since 5 is an odd number, the procedure prints 10. Since 10 is an even number, the procedure prints 5. Since 5 is an odd number, the procedure prints 10. Since 10 is an even number, the procedure prints 5. Since 5 is an odd number, the procedure prints 10. Since 10 is an even number, the procedure prints 5. Since 5 is an odd number, the procedure prints 10. Since 10 is an even number, the procedure prints 5. Since 5 is an odd number, the procedure prints 10. Since 10 is an even number, the procedure prints 5. Since 5 is an odd number, the procedure prints 10. Since 10 is an even number, the procedure prints 5. Since 5 is an odd number, the procedure prints 10. Since 10 is an even number, the procedure prints 5. Since 5 is an odd number, the procedure prints 10. Since 10 is an even number, the procedure prints 5. Since 5 is an odd number, the procedure prints 10. Since 10 is an even number, the procedure prints 5. Since 5 is an odd number, the procedure prints 10. Since 10 is an even number, the procedure prints 5. Since 5 is an odd number, the procedure prints 10. Since 10 is an even number, the procedure prints 5. Since 5 is an odd number, the procedure prints 10. Since 10 is an even number, the procedure prints 5. Since 5 is an odd number, the procedure prints 10. Since 10 is an even number, the procedure prints 5. Since 5 is an odd number, the procedure prints 10.

13. The correct answer is (D). The Boolean expression \(( \text{age} > 10 \) \( \land \) \( \text{height} > 36 \)) is true exactly when both \( \text{age} \) is greater than 10 and \( \text{height} \) is greater than 36. The same Boolean expression is false exactly either when \( \text{age} \) is less than or equal to 10 or when \( \text{height} \) is less than or equal to 36, which is the argument of \( \lnot \) () in choice (D). Another approach to solve this question is to use Boolean logic rules. According to one of De Morgan’s laws, if \( p \) and \( q \) represent Boolean variables, the Boolean expression \( p \text{ and } q \) is equivalent to the Boolean expression \( \lnot ( \lnot ( p ) \lor \lnot ( q ) ) \). If \( p \) represents the Boolean expression \( \text{age} > 10 \) and if \( q \) represents the Boolean expression \( \text{height} > 36 \), then \( \lnot ( p ) \) represents \( \text{age} \leq 10 \) and \( \lnot ( q ) \) represents \( \text{height} \leq 36 \).

14. The correct answer is (A). To transform ”Applesauce” into ”aPPLESAUCE”, the character at position 0 needs to change to lowercase and the characters from position 1 through the end of the string need to change to uppercase. The string that represents the character at position 0 is value.substring (0, 1 ). The string that represents the characters at positions 1 and higher is value.substring (1, value.length()).

15. The correct answer is (D). The for loop iterates over the values of \( c \) from 1 to \( n \), but the value of \( \text{temp} \) increases by \( c \) only when \( c \) is a multiple of 3.

16. The correct answer is (C). In scenario 1, the cars are washed in first-in first-out order, which characterizes the functionality of a queue. In scenario 2, the unique numeric IDs represent a mapping function, which is best supported by a dictionary/map data structure. In scenario 3, the balls are inserted in and removed from the can in first-in last-out order, which characterizes the functionality of a stack.

17. The correct answer is (B). Since 20 is greater than 15, the outer if condition evaluates to true and then the condition \( a > c \) is evaluated. Since 20 is not greater than 25, the condition \( a > c \) evaluates to false and \( \text{max} \) is not updated. The result is that the value of \( \text{max} \) at the end of the code segment is -1 instead of 25.

18. The correct answer is (C). Huffman coding is appropriately used for lossless compression because the original values can be recovered from the uniquely assigned codes with no loss of data.

19. The correct answer is (D). A function that sorts values in a column or row will facilitate detecting extreme values in that column or row.

20. The correct answer is (B). The model is validated by comparing the simulation results with the real-world data.

21. The correct answer is (C). The size of the storage needed is approximately \( 3 \times 10^2 \times 4 \times 10^2 \times 5 \times 10^3 \times 10^3 \times 5 \times 1 \) bytes, which is equivalent to \( 3 \times 10^{12} \) bytes. This number is best expressed in terabytes.

22. The correct answer is (C). An Internet-connected smart refrigerator that can track its milk inventory would be able to communicate with online delivery services and order milk shipments whenever the milk inventory is low.

23. The correct answer is (D). A compiler translates a program written in a high-level programming language into low-level instructions that can be read and executed by the computer.

24. The correct answer is (A). A Web server is a server that provides content to clients by using Hypertext Transfer Protocol (HTTP).

25. The correct answer is (A). Authentication is the process of confirming a valid identification. Successful use of false credentials results in incorrect identification of the user.
4. Determine Your Strategy for Success

*Set clear goals and deadlines so your test preparation is focused and efficient*

Effective Praxis test preparation doesn't just happen. You'll want to set clear goals and deadlines for yourself along the way. Otherwise, you may not feel ready and confident on test day.

1) **Learn what the test covers.**

You may have heard that there are several different versions of the same test. It's true. You may take one version of the test and your friend may take a different version a few months later. Each test has different questions covering the same subject area, but both versions of the test measure the same skills and content knowledge.

You'll find specific information on the test you're taking on page 5, which outlines the content categories that the test measures and what percentage of the test covers each topic. Visit [www.ets.org/praxis/testprep](http://www.ets.org/praxis/testprep) for information on other Praxis tests.

2) **Assess how well you know the content.**

Research shows that test takers tend to overestimate their preparedness—this is why some test takers assume they did well and then find out they did not pass.

The Praxis tests are demanding enough to require serious review of likely content, and the longer you've been away from the content, the more preparation you will most likely need. If it has been longer than a few months since you've studied your content area, make a concerted effort to prepare.

3) **Collect study materials.**

Gathering and organizing your materials for review are critical steps in preparing for the Praxis tests. Consider the following reference sources as you plan your study:

- Did you take a course in which the content area was covered? If yes, do you still have your books or your notes?
- Does your local library have a high school-level textbook in this area? Does your college library have a good introductory college-level textbook in this area?

Practice materials are available for purchase for many Praxis tests at [www.ets.org/praxis/testprep](http://www.ets.org/praxis/testprep). Test preparation materials include sample questions and answers with explanations.

4) **Plan and organize your time.**

You can begin to plan and organize your time while you are still collecting materials. Allow yourself plenty of review time to avoid cramming new material at the end. Here are a few tips:

- Choose a test date far enough in the future to leave you plenty of preparation time. Test dates can be found at [http://www.ets.org/praxis/register/dates_centers](http://www.ets.org/praxis/register/dates_centers).
- Work backward from that date to figure out how much time you will need for review.
- Set a realistic schedule—and stick to it.
5) Practice explaining the key concepts.

*Praxis* tests with constructed-response questions assess your ability to explain material effectively. As a teacher, you'll need to be able to explain concepts and processes to students in a clear, understandable way. What are the major concepts you will be required to teach? Can you explain them in your own words accurately, completely, and clearly? Practice explaining these concepts to test your ability to effectively explain what you know.

6) Understand how questions will be scored.

Scoring information can be found on page 55.

7) Develop a study plan.

A study plan provides a road map to prepare for the *Praxis* tests. It can help you understand what skills and knowledge are covered on the test and where to focus your attention. Use the study plan template on page 37 to organize your efforts.

And most important—get started!

Would a Study Group Work for You?

Using this guide as part of a study group

People who have a lot of studying to do sometimes find it helpful to form a study group with others who are working toward the same goal. Study groups give members opportunities to ask questions and get detailed answers. In a group, some members usually have a better understanding of certain topics, while others in the group may be better at other topics. As members take turns explaining concepts to one another, everyone builds self-confidence.

If the group encounters a question that none of the members can answer well, the group can go to a teacher or other expert and get answers efficiently. Because study groups schedule regular meetings, members study in a more disciplined fashion. They also gain emotional support. The group should be large enough so that multiple people can contribute different kinds of knowledge, but small enough so that it stays focused. Often, three to six members is a good size.

Here are some ways to use this guide as part of a study group:

- **Plan the group’s study program.** Parts of the study plan template, beginning on page 37 can help to structure your group’s study program. By filling out the first five columns and sharing the worksheets, everyone will learn more about your group’s mix of abilities and about the resources, such as textbooks, that members can share with the group. In the sixth column (“Dates I will study the content”), you can create an overall schedule for your group’s study program.

- **Plan individual group sessions.** At the end of each session, the group should decide what specific topics will be covered at the next meeting and who will present each topic. Use the topic headings and subheadings in the Test at a Glance table on page 5 to select topics, and then select practice questions, beginning on page 19.

- **Prepare your presentation for the group.** When it’s your turn to present, prepare something that is more than a lecture. Write two or three original questions to pose to the group. Practicing writing actual questions can help you better understand the topics covered on the test as well as the types of questions you will encounter on the test. It will also give other members of the group extra practice at answering questions.
Step 4: Determine Your Strategy for Success

• **Take a practice test together.** The idea of a practice test is to simulate an actual administration of the test, so scheduling a test session with the group will add to the realism and may also help boost everyone's confidence. Remember, complete the practice test using only the time that will be allotted for that test on your administration day.

• **Learn from the results of the practice test.** Review the results of the practice test, including the number of questions answered correctly in each content category. For tests that contain constructed-response questions, look at the Sample Test Questions section, which also contain sample responses to those questions and shows how they were scored. Then try to follow the same guidelines that the test scorers use.

• **Be as critical as you can.** You’re not doing your study partner(s) any favors by letting them get away with an answer that does not cover all parts of the question adequately.

• **Be specific.** Write comments that are as detailed as the comments about the sample responses. Indicate where and how your study partner(s) are doing an inadequate job of answering the question. Writing notes in the margins of the answer sheet may also help.

• **Be supportive.** Include comments that point out what your study partner(s) got right.

Then plan one or more study sessions based on aspects of the questions on which group members performed poorly. For example, each group member might be responsible for rewriting one paragraph of a response in which someone else did an inadequate job.

Whether you decide to study alone or with a group, remember that the best way to prepare is to have an organized plan. The plan should set goals based on specific topics and skills that you need to learn, and it should commit you to a realistic set of deadlines for meeting those goals. Then you need to discipline yourself to stick with your plan and accomplish your goals on schedule.
## 5. Develop Your Study Plan

### Develop a personalized study plan and schedule

Planning your study time is important because it will help ensure that you review all content areas covered on the test. Use the sample study plan below as a guide. It shows a plan for the Core Academic Skills for Educators: Reading test. Following that is a study plan template that you can fill out to create your own plan. Use the “Learn about Your Test” and “Test Specifications” information beginning on page 5 to help complete it.

### Use this worksheet to:
1. Define Content Areas: List the most important content areas for your test as defined in chapter 1.
2. Determine Strengths and Weaknesses: Identify your strengths and weaknesses in each content area.
3. Identify Resources: Identify the books, courses, and other resources you plan to use for each content area.
4. Study: Create and commit to a schedule that provides for regular study periods.

| Praxis Test Name (Test Code): | Core Academic Skills for Educators: Reading (5712) |
| Test Date: | 9/15/17 |

<table>
<thead>
<tr>
<th>Content covered</th>
<th>Description of content</th>
<th>How well do I know the content? (scale 1–5)</th>
<th>What resources do I have/need for the content?</th>
<th>Where can I find the resources I need?</th>
<th>Dates I will study the content</th>
<th>Date completed</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Key Ideas and Details</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Close reading</td>
<td>Draw inferences and implications from the directly stated content of a reading selection</td>
<td>3</td>
<td>Middle school English textbook</td>
<td>College library, middle school teacher</td>
<td>7/15/17</td>
<td>7/15/17</td>
</tr>
<tr>
<td>Determining Ideas</td>
<td>Identify summaries or paraphrases of the main idea or primary purpose of a reading selection</td>
<td>3</td>
<td>Middle school English textbook</td>
<td>College library, middle school teacher</td>
<td>7/17/17</td>
<td>7/17/17</td>
</tr>
<tr>
<td>Determining Ideas</td>
<td>Identify summaries or paraphrases of the supporting ideas and specific details in a reading selection</td>
<td>3</td>
<td>Middle and high school English textbook</td>
<td>College library, middle and high school teachers</td>
<td>7/20/17</td>
<td>7/21/17</td>
</tr>
<tr>
<td><strong>Craft, Structure, and Language Skills</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Interpreting tone</td>
<td>Determine the author’s attitude toward material discussed in a reading selection</td>
<td>4</td>
<td>Middle and high school English textbook</td>
<td>College library, middle and high school teachers</td>
<td>7/25/17</td>
<td>7/26/17</td>
</tr>
<tr>
<td>Analysis of structure</td>
<td>Identify key transition words and phrases in a reading selection and how they are used</td>
<td>3</td>
<td>Middle and high school English textbook, dictionary</td>
<td>College library, middle and high school teachers</td>
<td>7/25/17</td>
<td>7/27/17</td>
</tr>
<tr>
<td>Analysis of structure</td>
<td>Identify how a reading selection is organized in terms of cause/effect, compare/contrast, problem/solution, etc.</td>
<td>5</td>
<td>High school textbook, college course notes</td>
<td>College library, course notes, high school teacher, college professor</td>
<td>8/1/17</td>
<td>8/1/17</td>
</tr>
<tr>
<td>Author’s purpose</td>
<td>Determine the role that an idea, reference, or piece of information plays in an author’s discussion or argument</td>
<td>5</td>
<td>High school textbook, college course notes</td>
<td>College library, course notes, high school teacher, college professor</td>
<td>8/1/17</td>
<td>8/1/17</td>
</tr>
</tbody>
</table>

(continued on next page)
## Step 5: Develop Your Study Plan

<table>
<thead>
<tr>
<th>Content covered</th>
<th>Description of content</th>
<th>How well do I know the content? (scale 1–5)</th>
<th>What resources do I have/need for the content?</th>
<th>Where can I find the resources I need?</th>
<th>Dates I will study the content</th>
<th>Date completed</th>
</tr>
</thead>
<tbody>
<tr>
<td>Language in different contexts</td>
<td>Determine whether information presented in a reading selection is presented as fact or opinion</td>
<td>4</td>
<td>High school textbook, college course notes</td>
<td>College library, course notes, high school teacher, college professor</td>
<td>8/1/17</td>
<td>8/1/17</td>
</tr>
<tr>
<td>Contextual meaning</td>
<td>Identify the meanings of words as they are used in the context of a reading selection</td>
<td>2</td>
<td>High school textbook, college course notes</td>
<td>College library, course notes, high school teacher, college professor</td>
<td>8/1/17</td>
<td>8/1/17</td>
</tr>
<tr>
<td>Figurative Language</td>
<td>Understand figurative language and nuances in word meanings</td>
<td>2</td>
<td>High school textbook, college course notes</td>
<td>College library, course notes, high school teacher, college professor</td>
<td>8/8/17</td>
<td>8/8/17</td>
</tr>
<tr>
<td>Vocabulary range</td>
<td>Understand a range of words and phrases sufficient for reading at the college and career readiness level</td>
<td>2</td>
<td>High school textbook, college course notes</td>
<td>College library, course notes, high school teacher, college professor</td>
<td>8/15/17</td>
<td>8/17/17</td>
</tr>
</tbody>
</table>

### Integration of Knowledge and Ideas

<p>| | | | | | | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Diverse media and formats</td>
<td>Analyze content presented in diverse media and formats, including visually and quantitatively, as well as in words</td>
<td>2</td>
<td>High school textbook, college course notes</td>
<td>College library, course notes, high school teacher, college professor</td>
<td>8/22/17</td>
<td>8/24/17</td>
</tr>
<tr>
<td>Evaluation of arguments</td>
<td>Identify the relationship among ideas presented in a reading selection</td>
<td>4</td>
<td>High school textbook, college course notes</td>
<td>College library, course notes, high school teacher, college professor</td>
<td>8/24/17</td>
<td>8/24/17</td>
</tr>
<tr>
<td>Evaluation of arguments</td>
<td>Determine whether evidence strengthens, weakens, or is relevant to the arguments in a reading selection</td>
<td>3</td>
<td>High school textbook, college course notes</td>
<td>College library, course notes, high school teacher, college professor</td>
<td>8/27/17</td>
<td>8/27/17</td>
</tr>
<tr>
<td>Evaluation of arguments</td>
<td>Determine the logical assumptions upon which an argument or conclusion is based</td>
<td>5</td>
<td>High school textbook, college course notes</td>
<td>College library, course notes, high school teacher, college professor</td>
<td>8/28/17</td>
<td>8/30/17</td>
</tr>
<tr>
<td>Evaluation of arguments</td>
<td>Draw conclusions from material presented in a reading selection</td>
<td>5</td>
<td>High school textbook, college course notes</td>
<td>College library, course notes, high school teacher, college professor</td>
<td>8/30/17</td>
<td>8/31/17</td>
</tr>
<tr>
<td>Comparison of texts</td>
<td>Recognize or predict ideas or situations that are extensions of or similar to what has been presented in a reading selection</td>
<td>4</td>
<td>High school textbook, college course notes</td>
<td>College library, course notes, high school teacher, college professor</td>
<td>9/3/17</td>
<td>9/4/17</td>
</tr>
<tr>
<td>Comparison of texts</td>
<td>Apply ideas presented in a reading selection to other situations</td>
<td>2</td>
<td>High school textbook, college course notes</td>
<td>College library, course notes, high school teacher, college professor</td>
<td>9/5/17</td>
<td>9/6/17</td>
</tr>
</tbody>
</table>
## My Study Plan

Use this worksheet to:

1. **Define Content Areas:** List the most important content areas for your test as defined in chapter 1.
2. **Determine Strengths and Weaknesses:** Identify your strengths and weaknesses in each content area.
3. **Identify Resources:** Identify the books, courses, and other resources you plan to use for each content area.
4. **Study:** Create and commit to a schedule that provides for regular study periods.

### Praxis Test Name (Test Code):

____________________________________________________________

### Test Date:

__________

<table>
<thead>
<tr>
<th>Content covered</th>
<th>Description of content</th>
<th>How well do I know the content? (scale 1–5)</th>
<th>What resources do I have/need for this content?</th>
<th>Where can I find the resources I need?</th>
<th>Dates I will study this content</th>
<th>Date completed</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

(continued on next page)
<table>
<thead>
<tr>
<th>Content covered</th>
<th>Description of content</th>
<th>How well do I know the content? (scale 1–5)</th>
<th>What resources do I have/need for the content?</th>
<th>Where can I find the resources I need?</th>
<th>Dates I will study the content</th>
<th>Date completed</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
6. Review Study Topics

Detailed study topics with questions for discussion

Using the Study Topics That Follow

The Computer Science test is designed to measure the knowledge and skills necessary for a beginning teacher. This chapter is intended to help you organize your preparation for the test and to give you a clear indication of the depth and breadth of the knowledge required for success on the test.

Virtually all accredited programs address the topics covered by the test; however, you are not expected to be an expert on all aspects of the topics that follow.

You are likely to find that the topics below are covered by most introductory textbooks. Consult materials and resources, including lecture and laboratory notes, from all your course work. You should be able to match up specific topics and subtopics with what you have covered in your courses.

Try not to be overwhelmed by the volume and scope of content knowledge in this guide. Although a specific term may not seem familiar as you see it here, you might find that you can understand it when applied to a real-life situation. Many of the items on the actual test will provide you with a context for these topics or terms.

Discussion Areas

Interspersed throughout the study topics are discussion areas, presented as open-ended questions or statements that are displayed in bulleted lists. These discussion areas are intended to help test your knowledge of fundamental concepts and your ability to apply those concepts to situations in the classroom or the real world. Most of the areas require you to combine several pieces of knowledge to formulate an integrated understanding and response. If you spend time on these areas, you will gain increased understanding and facility with the subject matter covered on the test. You might want to discuss these areas and your answers with a teacher or mentor.

Note that this study companion does not provide answers for the discussion area questions, but thinking about the answers to them will help improve your understanding of fundamental concepts and will probably help you answer a broad range of questions on the test.
Study Topics

An overview of the areas covered on the test, along with their subareas, follows.

I. Impacts of Computing

A. Understands and applies knowledge of impact of, obstacles to, and effects of computing

1. Understand computing as a way of expressing creativity, solving problems, enabling communication, and fostering innovation in a variety of fields and careers
   a. recognize that computers can be used to showcase creativity
   b. recognize the benefits of using computers to solve problems
   c. provide examples of how computers enable communication and collaboration
   d. provide examples of how computers foster innovation

2. Know the obstacles to equal access to computing among different groups and the impact of those obstacles
   a. identify obstacles to equal access to computing among different groups (e.g., groups defined by gender, socioeconomic status, disability/accessibility needs) and the impact of those obstacles
   b. identify factors that contribute to the digital divide
   c. match obstacles to equal access with effective solutions

3. Understand beneficial and harmful effects of computing innovations and the trade-offs between them
   a. analyze computing innovations in terms of their social, economic, and cultural impacts, both beneficial and harmful
   b. identify trade-offs between beneficial and harmful effects of computer innovations

B. Understands and applies knowledge of issues regarding intellectual property, ethics, privacy, and security in computing

1. Know different methods of protecting intellectual property rights and the trade-offs between them in a variety of contexts (e.g., Creative Commons, open source, copyright)
   a. using correct vocabulary, describe how different methods of protecting intellectual property rights work
   b. given a context, identify appropriate methods of protecting intellectual property rights
   c. identify and compare trade-offs between different methods of protecting intellectual property rights

2. Understand ethical and unethical computing practices and their social, economic, and cultural implications
   a. identify ethical and unethical computing practices in context
   b. describe the social, economic, and cultural implications of ethical and unethical computing practices
   c. identify the conditions under which a given computing practice is ethical or legal

3. Know privacy and security issues regarding the acquisition, use, and disclosure of information in a digital world
   a. using correct vocabulary, describe privacy and security issues
   b. in context, identify appropriate strategies to safeguard privacy and ensure security
   c. describe trade-offs between local and cloud-based data storage
   d. identify methods that digital services use to collect information about users

Discussion areas: Impacts of Computing

• Can you give examples of computing innovations?
• Can you describe ways that computing enables creativity and innovation?
• Can you give examples of computing innovations that have impacted the quality of life?
• Can you give examples of how computers enable communication and collaboration in your life?

• Can you give examples of how computers foster innovation?

• Can you recognize obstacles to equal access to computing among different groups (e.g., groups defined by gender, socioeconomic status, disability/accessibility needs)?

• Can you give examples of effective solutions that address equal access for these groups?

• Can you describe what is meant by the digital divide? Can you give examples?

• Given a computing innovation, can you identify beneficial impacts as well as harmful impacts?

• Can you explain the value of copyrights? How does copyright affect the use of teaching and learning materials in your classroom?

• Can you describe what open source means? Can you give examples of commonly used open source applications?

• Can you give examples of materials that have a Creative Commons license?

• Can you describe how Creative Commons licenses are different from other types of copyright licenses?

• Can you give examples of ethical and unethical computing practices?

• Can you describe the social, economic, and cultural implications of ethical and unethical computing practices?

• Can you describe strategies that are used to safeguard privacy?

• Can you describe strategies that are used to ensure security?

• Can you give examples of data that is stored locally? Can you give examples of data that is stored in the cloud? Can you describe advantages and disadvantages of local and cloud-based data storage?

• Can you describe what types of information digital service providers collect about their users? How does that influence practices in your classroom?

II. Algorithms and Computational Thinking

A. Understands and applies knowledge of abstraction, pattern recognition, problem decomposition, number base conversion, and algorithm formats

1. Understand abstraction as a foundation of computer science
   a. identify, create, or complete the correct ordering, from low to high, of an abstraction hierarchy
   b. identify abstractions in context
   c. identify details that can be removed from a solution in order to generalize it

2. Know how to use pattern recognition, problem decomposition, and abstraction to develop an algorithm
   a. given a table of values or other data source, identify the patterns in the data and identify algorithms that could produce the patterns
   b. identify components that could be part of an algorithm to solve a problem
   c. identify actions and actors when decomposing a problem
   d. identify appropriate decomposition strategies

3. Understand number base conversion and binary, decimal, and hexadecimal number systems
   a. convert between number bases
   b. analyze and compare representations of numbers in different bases

4. Understand how to develop and analyze algorithms expressed in multiple formats (e.g., natural language, flowcharts, pseudocode)
   a. interpret diagrams that describe algorithms, given an explanation of the symbols used
   b. compare algorithms written in multiple formats
   c. trace and analyze algorithms written in different formats
   d. identify correct sequencing of steps in an algorithm and errors in sequencing
B. **Understands and applies knowledge of algorithm analysis, searching and sorting algorithms, recursive algorithms, and randomization**

1. Be familiar with the limitations of computing in terms of time, space, and solvability as well as with the use of heuristic solutions that can address these limitations
   a. identify and compare algorithms that are linear, quadratic, exponential, or logarithmic
   b. recognize the existence of problems that cannot be solved by a computer
   c. in context, identify factors that prevent a problem from being solvable
   d. identify situations where heuristic solutions are useful
   e. in context, identify space and time limitations of computational solutions to problems

2. Understand searching and sorting algorithms; can analyze sorting algorithms for correctness and can analyze searching algorithms for correctness and efficiency
   a. trace algorithms and predict output and intermediate results
   b. calculate the number of comparisons required for linear and binary search algorithms

3. Understand simple recursive algorithms (e.g., n factorial, sum of first n integers)
   a. trace simple recursive algorithms
   b. provide missing steps in incomplete simple recursive algorithms
   c. identify parts of a recursive algorithm (e.g., base or stopping condition, recursive call)
   d. identify errors in simple recursive algorithms
   e. identify an iterative algorithm that is equivalent to a recursive algorithm

4. Be familiar with the use of randomization in computing
   a. identify appropriate uses of randomization in a variety of applications
   b. identify the difference between random and pseudorandom numbers

---

**Discussion areas: Algorithms and Computational Thinking**

- Can you give examples of various levels of abstractions?
- Can you describe how abstractions such as numbers, characters, and colors are represented as bits?
- Given a data source, such as a table of values, can you describe an algorithm that would produce those patterns of data?
- Can you apply decomposition strategies to decompose a problem?
- Can you convert between decimal numbers, binary numbers, octal numbers, and hexadecimal numbers?
- Can you trace an algorithm given in natural language, flowcharts, or pseudocode?
- Are you familiar with the pseudocode notation used in this test and described in section 1?
- Can you express an algorithm in natural language, in flowcharts, and in pseudocode, and can you translate from one format to another?
- Can you analyze an algorithm to identify steps that are missing or out of order?
- Are all problems solvable by an algorithm? Can you describe a problem that cannot be solved by a computer?
- Can you give examples of algorithms that are linear, quadratic, exponential, or logarithmic?
- Can you give examples of problems where heuristic solutions are useful?
- Can you describe common comparison-based sorting algorithms, such as insertion sort, selection sort, etc., and analyze the number of comparisons?
- Can you describe linear search and binary search, and analyze the number of comparisons?
- Can you describe a recursive algorithm?
- Can you trace a recursive algorithm, and analyze the number of recursive calls?
Step 6: Review Study Topics

- Can you translate between a recursive algorithm and an equivalent iterative algorithm?
- Can you provide examples of problems where the use of randomization is useful?
- Can you describe the difference between random and pseudorandom numbers?
- Can you transform a random number generator to fit a specific range?

III. Programming

A. Understands and applies knowledge of programming control structures, standard operators, variables, correctness, extensibility, modifiability, and reusability

1. Understand how to write and modify computer programs in a text-based programming language
   a. describe what a program does or be able to choose the code segment that correctly implements a given intended purpose
   b. identify missing code in a code segment with a stated intended purpose
   c. place statements in appropriate order to create a correct program
   d. identify how changing one part of a code segment will affect the output

2. Understand how to analyze computer programs in terms of correctness
   a. trace code and indicate the output printed or the value of variables after code segment execution
   b. indicate inputs that produce given outputs for a code segment
   c. describe what a program does or choose the code segment that correctly implements a given intended purpose
   d. identify valid preconditions and postconditions
   e. compare two code segments or algorithms
   f. identify the type of error produced by a code segment (i.e., syntax, runtime, compile-time, overflow, round-off, logic)
   g. identify errors in incorrect code and changes that can be made to correct them

3. Know the concepts of extensibility, modifiability, and reusability
   a. identify the meaning of the terms
   b. identify functionally equivalent statements or code segments that differ in one of these three ways
   c. identify situations where the use of constants or variables would be preferred over hard-coded values
   d. identify opportunities for parameterization
   e. choose code that improves on given code by making it more extensible, modifiable, or reusable
   f. identify changes that would improve a given code segment

4. Understand the three basic constructs used in programming: sequence, selection, and iteration
   a. trace code and indicate the output printed or the value of variables after code segment execution
   b. indicate inputs that produce given outputs for a code segment
   c. describe what a program does or choose the code segment that correctly implements a given intended purpose
   d. identify missing code in a code segment with a stated intended purpose
   e. identify equivalent statements or code segments
   f. identify the three constructs when used in code
   g. identify which of the constructs are needed to implement given functionality
   h. convert code that does not use iteration to equivalent code that uses iteration

5. Understand how to use standard operators (i.e., assignment, arithmetic, relational, logical) and operator precedence to write programs
   a. trace code and indicate the output displayed or the value of variables after code segment execution
   b. indicate inputs that produce given outputs for a code segment
   c. describe what a program does or choose the code segment that correctly implements a stated intended purpose
   d. identify missing code in a code segment with a stated intended purpose
Step 6: Review Study Topics

- identify equivalent statements or code segments
- place statements in appropriate order to create a correct program
- use Boolean algebra to identify equivalent Boolean expressions
- write a Boolean expression equivalent to given code, or identify code equivalent to a given Boolean expression or English description
- identify the correct implementation of a given formula, including formulas with fractions
- evaluate expressions that include arithmetic operations

6. Understand how to use variables and a variety of data types
   - identify variables and data types (e.g., integers, floating point, string, Booleans, arrays/lists)
   - identify the need for type conversion
   - trace code and indicate the output printed or the value of variables after code segment execution
   - indicate the inputs that produce given outputs for a code segment
   - describe what a program does or choose the code segment that correctly implements a stated intended purpose
   - identify missing code in a code segment with a stated intended purpose
   - identify equivalent statements or code segments
   - place statements in appropriate order to create a correct program
   - describe the difference between integer and floating point numeric data types
   - describe the difference between integer and floating point division
   - describe the benefits of the use of each data type
   - distinguish between global and local scope
   - identify the most appropriate data type in a given context
   - identify the correct sequence of string operations to produce a given output

B. Understands and applies knowledge of procedures, event-driven programs, usability, data structures, debugging, documenting and reviewing code, libraries and APIs, IDEs, and programming language paradigms, including object-oriented concepts

1. Understand how to write and call procedures with parameters and return values
   - trace code and indicate the output printed or the value of variables after code segment execution
   - indicate inputs that produce given outputs for a code segment
   - describe what a program does or choose the code segment that correctly implements a stated intended purpose
   - identify missing code in a code segment with a stated intended purpose
   - identify equivalent statements or code segments
   - place statements in appropriate order to create a correct program
   - trace code when references to objects and arrays are passed to procedures
   - trace code that includes nested procedure calls

2. Know the concepts of event-driven programs that respond to external events (e.g., sensors, messages, clicks)
   - trace code and indicate the output printed or the value of variables after code segment execution
   - indicate inputs that produce given outputs for a code segment
   - describe what a program does or choose the code segment that correctly implements a stated intended purpose
   - identify missing code in a code segment with a stated intended purpose
   - identify possible errors due to asynchronous events
   - identify aspects of concurrency in event-driven programming

3. Be familiar with usability and user experience (e.g., ease of use and accessibility)
   - identify code that improves on given code in terms of usability or user experience
Step 6: Review Study Topics

4. Be familiar with dictionaries/maps, stacks, and queues
   a. identify a data structure based on a description of behavior or appropriate use
   b. given goals, constraints, or context, identify the most appropriate data structure
   c. trace code that uses a particular data structure

5. Understand how to use debugging techniques and appropriate test cases
   a. identify which test cases are most useful for given code
   b. differentiate between different types of errors (e.g., overflow, round-off, syntax, runtime, compile-time, logic)
   c. describe useful debugging techniques (e.g., where to put print statements)
   d. differentiate between empirical testing and proof
   e. identify errors in code and solutions to those errors

6. Be familiar with characteristics of well-documented computer programs that are usable, readable, and modular
   a. identify characteristics of good documentation
   b. identify good and poor documentation practices in context

7. Be familiar with techniques to obtain and use feedback to produce high-quality code (e.g., code reviews, peer feedback, end user feedback)
   a. identify situations in which each of the three listed techniques are useful

8. Know how to use libraries and APIs
   a. identify correct call(s) and use of return values given an API definition
   b. identify reasons to use or not use libraries in place of writing original code
   c. identify applications (e.g., math libraries, random number generation) that use APIs

9. Understand programming techniques to validate correct input and detect incorrect input
   a. identify effective input data validation strategies

b. compare data validation (proper range and format) and data verification (e.g., password verification)

10. Be familiar with the features and capabilities of integrated development environments (IDEs)
    a. identify components of IDEs
    b. identify benefits and drawbacks of using IDEs
    c. identify the costs and benefits of context editors

11. Be familiar with the differences between low- and high-level programming languages
    a. identify characteristics of low- and high-level languages

12. Be familiar with different programming paradigms
    a. identify the terminology of procedural programming
    b. identify the terminology of object-oriented programming
    c. compare programming paradigms

13. Know object-oriented programming concepts
    a. identify classes, instance variables, and methods given a diagram
    b. identify the benefits of inheritance and encapsulation
    c. identify distinctions between overloading and overriding

14. Be familiar with program compilation and program interpretation
    a. identify differences between compilation and interpretation
    b. identify differences between source code and object code

Discussion areas: Programming

• Can you analyze a pseudocode program and trace its execution?
• Given a pseudocode program, can you describe what it does?
• Given an incomplete pseudocode program, can you identify the missing code?
• Can you analyze a pseudocode program in terms of correctness?
Step 6: Review Study Topics

- Can you give examples of different types of errors (syntax, run-time, compile-time, overflow, round-off, logic) and describe how to fix the errors?
- Given a pseudocode program, can you identify valid preconditions and postconditions?
- Can you describe the concepts of extensibility, modifiability, and reusability?
- Given a pseudocode program, can you describe how to improve it by making it more extensible, modifiable, or reusable?
- Can you give examples where the use of constants or variables would be preferred over hard-coded values?
- Can you trace code that uses loops (for, while, do-while, etc.) and convert from one type of loop to another?
- Can you trace code that uses selection (if, if/else, switch, case, etc.) and convert from one type of selection to another?
- Can you evaluate expressions that use standard operators (assignment, arithmetic, relational, logical) and operator precedence?
- Can you use the rules of Boolean algebra to identify equivalent Boolean expressions?
- Can you describe the difference between decimal division and integer division?
- Can you describe the standard data types (integers, floating point, string, Booleans, arrays/lists)?
- Can you convert from one data type to another?
- Can you describe how the standard data types are stored in memory?
- Can you describe the difference between integer and floating point numeric data types?
- Can you evaluate an expression that uses string operations?
- Given a pseudocode program, can you determine the scope of a variable?
- Can you trace code that uses procedures with parameters and return values?
- Can you trace code that includes nested procedure calls?
- Can you analyze programs that respond to external events (sensors, messages, clicks)?
- Can you describe possible issues caused by concurrency?
- Can you describe how you have addressed usability and user experience in previous projects?
- Can you trace code that uses dictionaries/maps, stacks, and queues?
- Can you give an example where the use of a dictionary/map data structure is appropriate? Can you give examples for stacks and queues?
- Can you describe debugging techniques you have used in previous projects?
- Can you describe how you selected test cases when testing code in previous projects?
- Can you describe good and poor documentation practices?
- Can you give examples of how you used code reviews, peer feedback, or end user feedback in your past projects?
- Can you describe how you worked with libraries and API in previous projects?
- Can you describe what input data validation techniques you used in previous projects?
- Can you describe features and capabilities of IDEs?
- Can you give examples of low-level and high-level programming languages?
- Can you describe differences between different programming paradigms (procedural, object-oriented, etc.)?
- Can you give an example of overloading and an example of overriding, and explain the difference between overloading and overriding?
IV. Data

A. Understands and applies knowledge of digitalization, data encryption and decryption, and computational tools

1. Understand bits as the universal medium for expressing digital information
   a. perform calculations, using bits and bytes
   b. determine the number of bits and bytes required to store a given amount of data
   c. given the description of an encoding scheme, encode or decode data
   d. describe lossy and lossless data compression
   e. explain why binary numbers are fundamental to the operation of computer systems

2. Be familiar with concepts of data encryption and decryption
   a. distinguish between encoding and encryption
   b. identify trade-offs in the use of data encryption

3. Know how to use computational tools, including spreadsheets, to analyze data in order to discover, explain, and visualize patterns, connections, and trends
   a. transform data to make it more useful
   b. identify specific data or characteristics of specific data that need to be removed or modified before an entire data set can be used
   c. describe the use of spreadsheet operations (e.g., formulas, filters, sorts, charts, graphs) to analyze and visualize data

B. Understands and applies knowledge of simulation, modeling, and manipulation of data

1. Be familiar with the use of computing in simulation and modeling
   a. describe questions that can be answered with a given simulation, or explain what data and process are required in a simulation in order to answer a given question
   b. trace code in a simulation context
   c. identify missing code in a simulation context

   d. identify the impact of changes to simulations (e.g., more or fewer variables, more or less data)
   e. identify applications of simulation and modeling

2. Be familiar with methods to store, manage, and manipulate data
   a. use terminology and concepts of files and databases
   b. identify measures of file size (e.g., byte, kilo, mega, giga, tera, peta)
   c. identify issues connected with the storage requirements of computing applications, including scale, redundancy, and backup

3. Be familiar with a variety of computational methods for data collection, aggregation, and generation
   a. identify the benefits of working with publicly available data sets
   b. identify the types of data generated by surveys and sensors
   c. identify examples of crowdsourcing and citizen science
   d. identify appropriate data-collection methods for a given context and purpose

Discussion areas: Data

- Can you describe how bits and binary numbers are fundamental elements of computer systems?
- Can you give examples of encoding and encryption, and explain the difference between them?
- Can you describe the difference between lossy and lossless data compression?
- Can you describe some steps that can be used to transform raw data to make it more useful or meaningful?
- Can you describe how spreadsheet operations can be used to transform and visualize data?
- Can you give examples of simulation and modeling?
- Can you analyze a simulation to identify missing code or suggest modifications?
Step 6: Review Study Topics

- Can you describe the relationships between the measures of file size (byte, kilo, mega, giga, tera, peta)?
- Can you describe the impact of redundancy, scale, and backup on storage requirements?
- Can you describe methods used to collect, aggregate, and generate data?
- Can you give examples of crowdsourcing and citizen science?
- Can you identify situations where simulation and modeling will be useful?
- Can you determine the storage size needed for a given dataset?
- Can you transform binary data to other forms such as decimal and hexadecimal?
- Can you identify the missing or erroneous step(s) in a simulation?
- Can you identify potential issues during data processing/manipulation?
- Can you determine what data can be generated from a given simulation?

V. Computing Systems and Networks

A. Understands and applies knowledge of operating systems, computing systems, communication between devices, and cloud computing

1. Know that operating systems are programs that control and coordinate interactions between hardware and software components
   a. identify hardware components and their functions
   b. identify software components and their functions
   c. identify common operating systems tasks
   d. identify resource issues that have an impact on functionality

2. Be familiar with computing systems embedded in everyday objects (e.g., Internet of Things [IoT], ATMs, medical devices)
   a. describe what an embedded system is
   b. define what the IoT is and how it is used
   c. describe how sensors are used in embedded systems

B. Understands and applies knowledge of networks, including security issues and the Web

1. Know components of networks
   a. identify network hardware devices and their functions
   b. describe possible abstraction models of networks

2. Be familiar with factors that have an impact on network functionality
   a. define basic terminology (e.g., bandwidth, load, latency)
   b. estimate necessary bandwidth and data size for a given situation
   c. identify critical resources for a given situation
3. Be familiar with how Internet and Web protocols work
   a. describe the purpose of protocols and identify common Internet and Web protocols
   b. compare IPv4 and IPv6
   c. identify and describe the basic parts of a URL (e.g., protocol, subdomain, domain name, port, path)
   d. describe the hierarchical structure of names in the domain name system (DNS)
   e. describe the purpose and function of IP addressing
   f. identify how Internet protocols address reliability, redundancy, and error handling

4. Be familiar with digital and physical strategies for maintaining security
   a. identify characteristics of strong passwords (e.g., length, bits per character)
   b. identify digital and physical security strategies
   c. identify trade-offs in the use of security measures (e.g., encryption, decryption, digital signatures and certificates)

5. Be familiar with concepts of cybersecurity
   a. identify and define the five pillars of cybersecurity: confidentiality, integrity, availability, nonrepudiation, and authentication

6. Be familiar with the components that make up the Web (e.g., HTTP, HTML, browsers, servers, clients)
   a. identify the uses of markup languages
   b. identify the purposes of browsers, servers, and clients

Discussion areas: Computing Systems and Networks

• Can you describe common operating system tasks?
• Can you describe hardware and software components and their functions?
• Can you describe what is meant by the Internet of Things?
• Can you describe what happens during each step of the fetch-decode-execute cycle and put the steps in the correct order?
• Can you identify advantages and disadvantages of local, network, and cloud computing and storage?
• Can you identify and compare wireless communication systems?
• Can you describe, identify, and compare wired communication systems?
• Can you identify and compare network types?
• Can you identify network hardware devices and their functions?
• Can you describe what bandwidth, load, and latency mean?
• Can you describe how the Internet works?
• Can you describe digital and physical security strategies that you have used?
• Can you describe confidentiality, integrity, availability, nonrepudiation, and authentication?
• Can you describe components that make up the Web (HTTP, HTML, browsers, servers, clients, etc.)?
7. Review Smart Tips for Success

Follow test-taking tips developed by experts

Learn from the experts. Take advantage of the following answers to questions you may have and practical tips to help you navigate the Praxis test and make the best use of your time.

Should I guess?

Yes. Your score is based on the number of questions you answer correctly, with no penalty or subtraction for an incorrect answer. When you don’t know the answer to a question, try to eliminate any obviously wrong answers and then guess at the correct one. Try to pace yourself so that you have enough time to carefully consider every question.

Can I answer the questions in any order?

You can answer the questions in order or skip questions and come back to them later. If you skip a question, you can also mark it so that you can remember to return and answer it later. Remember that questions left unanswered are treated the same as questions answered incorrectly, so it is to your advantage to answer every question.

Are there trick questions on the test?

No. There are no hidden meanings or trick questions. All of the questions on the test ask about subject matter knowledge in a straightforward manner.

Are there answer patterns on the test?

No. You might have heard this myth: the answers on tests follow patterns. Another myth is that there will never be more than two questions in a row with the correct answer in the same position among the choices. Neither myth is true. Select the answer you think is correct based on your knowledge of the subject.

Can I write on the scratch paper I am given?

Yes. You can work out problems on the scratch paper, make notes to yourself, or write anything at all. Your scratch paper will be destroyed after you are finished with it, so use it in any way that is helpful to you. But make sure to select or enter your answers on the computer.

Smart Tips for Taking the Test

1. Skip the questions you find extremely difficult. Rather than trying to answer these on your first pass through the test, you may want to leave them blank and mark them so that you can return to them later. Pay attention to the time as you answer the rest of the questions on the test, and try to finish with 10 or 15 minutes remaining so that you can go back over the questions you left blank. Even if you don’t know the answer the second time you read the questions, see if you can narrow down the possible answers, and then guess. Your score is based on the number of right answers, so it is to your advantage to answer every question.
2. **Keep track of the time.** The on-screen clock will tell you how much time you have left. You will probably have plenty of time to answer all of the questions, but if you find yourself becoming bogged down, you might decide to move on and come back to any unanswered questions later.

3. **Read all of the possible answers before selecting one.** For questions that require you to select more than one answer, or to make another kind of selection, consider the most likely answers given what the question is asking. Then reread the question to be sure the answer(s) you have given really answer the question. Remember, a question that contains a phrase such as “Which of the following does NOT …” is asking for the one answer that is NOT a correct statement or conclusion.

4. **Check your answers.** If you have extra time left over at the end of the test, look over each question and make sure that you have answered it as you intended. Many test takers make careless mistakes that they could have corrected if they had checked their answers.

5. **Don’t worry about your score when you are taking the test.** No one is expected to answer all of the questions correctly. Your score on this test is not analogous to your score on the *GRE*® or other tests. It doesn't matter on the *Praxis* tests whether you score very high or barely pass. If you meet the minimum passing scores for your state and you meet the state’s other requirements for obtaining a teaching license, you will receive a license. In other words, what matters is meeting the minimum passing score. You can find passing scores for all states that use the *Praxis* tests at [http://www.ets.org/s/praxis/pdf/passing_scores.pdf](http://www.ets.org/s/praxis/pdf/passing_scores.pdf) or on the web site of the state for which you are seeking certification/licensure.

6. **Use your energy to take the test, not to get frustrated by it.** Getting frustrated only increases stress and decreases the likelihood that you will do your best. Highly qualified educators and test development professionals, all with backgrounds in teaching, worked diligently to make the test a fair and valid measure of your knowledge and skills. Your state painstakingly reviewed the test before adopting it as a licensure requirement. The best thing to do is concentrate on answering the questions.
8. Check on Testing Accommodations

See if you qualify for accommodations to take the Praxis test

What if English is not my primary language?

*Praxis* tests are given only in English. If your primary language is not English (PLNE), you may be eligible for extended testing time. For more details, visit www.ets.org/praxis/register/accommodations/plne.

What if I have a disability or other health-related need?

The following accommodations are available for *Praxis* test takers who meet the Americans with Disabilities Act (ADA) Amendments Act disability requirements:

- Extended testing time
- Additional rest breaks
- Separate testing room
- Writer/recorder of answers
- Test reader
- Sign language interpreter for spoken directions only
- Perkins Brailler
- Braille slate and stylus
- Printed copy of spoken directions
- Oral interpreter
- Audio test
- Braille test
- Large print test book
- Large print answer sheet
- Listening section omitted

For more information on these accommodations, visit www.ets.org/praxis/register/disabilities.

**Note:** Test takers who have health-related needs requiring them to bring equipment, beverages, or snacks into the testing room or to take extra or extended breaks must request these accommodations by following the procedures described in the *Bulletin Supplement for Test Takers with Disabilities or Health-Related Needs* (PDF), which can be found at http://www.ets.org/s/disabilities/pdf/bulletin_supplement_test_takers_with_disabilities_health_needs.pdf.

You can find additional information on available resources for test takers with disabilities or health-related needs at www.ets.org/disabilities.
9. Do Your Best on Test Day

Get ready for test day so you will be calm and confident

You followed your study plan. You prepared for the test. Now it’s time to prepare for test day.

Plan to end your review a day or two before the actual test date so you avoid cramming. Take a dry run to the test center so you’re sure of the route, traffic conditions, and parking. Most of all, you want to eliminate any unexpected factors that could distract you from your ultimate goal—passing the Praxis test!

On the day of the test, you should:

- be well rested
- wear comfortable clothes and dress in layers
- eat before you take the test
- bring an acceptable and valid photo identification with you
- bring an approved calculator only if one is specifically permitted for the test you are taking (see Calculator Use, at http://www.ets.org/praxis/test_day/policies/calculators)
- be prepared to stand in line to check in or to wait while other test takers check in

You can't control the testing situation, but you can control yourself. Stay calm. The supervisors are well trained and make every effort to provide uniform testing conditions, but don't let it bother you if the test doesn't start exactly on time. You will have the allotted amount of time once it does start.

You can think of preparing for this test as training for an athletic event. Once you've trained, prepared, and rested, give it everything you've got.

What items am I restricted from bringing into the test center?

You cannot bring into the test center personal items such as:

- handbags, knapsacks, or briefcases
- water bottles or canned or bottled beverages
- study materials, books, or notes
- pens, pencils, scrap paper, or calculators, unless specifically permitted for the test you are taking (see Calculator Use, at http://www.ets.org/praxis/test_day/policies/calculators)
- any electronic, photographic, recording, or listening devices

Personal items are not allowed in the testing room and will not be available to you during the test or during breaks. You may also be asked to empty your pockets. At some centers, you will be assigned a space to store your belongings, such as handbags and study materials. Some centers do not have secure storage space available, so please plan accordingly.

Test centers assume no responsibility for your personal items.
Step 9: Do Your Best on Test Day

If you have health-related needs requiring you to bring equipment, beverages or snacks into the testing room or to take extra or extended breaks, you need to request accommodations in advance. Procedures for requesting accommodations are described in the Bulletin Supplement for Test Takers with Disabilities or Health-related Needs (PDF).

Note: All cell phones, smart phones (e.g., Android® devices, iPhones®, etc.), and other electronic, photographic, recording, or listening devices are strictly prohibited from the test center. If you are seen with such a device, you will be dismissed from the test, your test scores will be canceled, and you will forfeit your test fees. If you are seen using such a device, the device will be confiscated and inspected. For more information on what you can bring to the test center, visit www.ets.org/praxis/test_day/bring.

Are You Ready?

Complete this checklist to determine whether you are ready to take your test.

☐ Do you know the testing requirements for the license or certification you are seeking in the state(s) where you plan to teach?

☐ Have you followed all of the test registration procedures?

☐ Do you know the topics that will be covered in each test you plan to take?

☐ Have you reviewed any textbooks, class notes, and course readings that relate to the topics covered?

☐ Do you know how long the test will take and the number of questions it contains?

☐ Have you considered how you will pace your work?

☐ Are you familiar with the types of questions for your test?

☐ Are you familiar with the recommended test-taking strategies?

☐ Have you practiced by working through the practice questions in this study companion or in a study guide or practice test?

☐ If constructed-response questions are part of your test, do you understand the scoring criteria for these questions?

☐ If you are repeating a Praxis test, have you analyzed your previous score report to determine areas where additional study and test preparation could be useful?

If you answered "yes" to the questions above, your preparation has paid off. Now take the Praxis test, do your best, pass it—and begin your teaching career!
10. Understand Your Scores

*Understand how tests are scored and how to interpret your test scores*

Of course, passing the *Praxis* test is important to you so you need to understand what your scores mean and what your state requirements are.

**What are the score requirements for my state?**

States, institutions, and associations that require the tests set their own passing scores. Visit [www.ets.org/praxis/states](http://www.ets.org/praxis/states) for the most up-to-date information.

**If I move to another state, will my new state accept my scores?**

The *Praxis* tests are part of a national testing program, meaning that they are required in many states for licensure. The advantage of a national program is that if you move to another state that also requires *Praxis* tests, you can transfer your scores. Each state has specific test requirements and passing scores, which you can find at [www.ets.org/praxis/states](http://www.ets.org/praxis/states).

**How do I know whether I passed the test?**

Your score report will include information on passing scores for the states you identified as recipients of your test results. If you test in a state with automatic score reporting, you will also receive passing score information for that state.

A list of states and their passing scores for each test are available online at [www.ets.org/praxis/states](http://www.ets.org/praxis/states).

**What your *Praxis* scores mean**

You received your score report. Now what does it mean? It’s important to interpret your score report correctly and to know what to do if you have questions about your scores.


To access *Understanding Your Praxis Scores*, a document that provides additional information on how to read your score report, visit [www.ets.org/praxis/scores/understand](http://www.ets.org/praxis/scores/understand).

**Put your scores in perspective**

Your score report indicates:

- Your score and whether you passed
- The range of possible scores
- The raw points available in each content category
- The range of the middle 50 percent of scores on the test

If you have taken the same *Praxis* test or other *Praxis* tests over the last 10 years, your score report also lists the highest score you earned on each test taken.
Content category scores and score interpretation

Questions on the Praxis tests are categorized by content. To help you in future study or in preparing to retake the test, your score report shows how many raw points you earned in each content category. Compare your "raw points earned" with the maximum points you could have earned ("raw points available"). The greater the difference, the greater the opportunity to improve your score by further study.

Score scale changes

ETS updates Praxis tests on a regular basis to ensure they accurately measure the knowledge and skills that are required for licensure. When tests are updated, the meaning of the score scale may change, so requirements may vary between the new and previous versions. All scores for previous, discontinued tests are valid and reportable for 10 years, provided that your state or licensing agency still accepts them.

These resources may also help you interpret your scores:

- Understanding Your Praxis Scores (PDF), found at www.ets.org/praxis/scores/understand
- The Praxis Passing Scores (PDF), found at www.ets.org/praxis/scores/understand
- State requirements, found at www.ets.org/praxis/states
Appendix: Other Questions You May Have

Here is some supplemental information that can give you a better understanding of the Praxis tests.

What do the Praxis tests measure?
The Praxis tests measure the specific knowledge and skills that beginning teachers need. The tests do not measure an individual’s disposition toward teaching or potential for success, nor do they measure your actual teaching ability. The assessments are designed to be comprehensive and inclusive but are limited to what can be covered in a finite number of questions and question types. Teaching requires many complex skills that are typically measured in other ways, including classroom observation, video recordings, and portfolios.

Ranging from Agriculture to World Languages, there are more than 80 Praxis tests, which contain selected-response questions or constructed-response questions, or a combination of both.

Who takes the tests and why?
Some colleges and universities use the Praxis Core Academic Skills for Educators tests (Reading, Writing, and Mathematics) to evaluate individuals for entry into teacher education programs. The assessments are generally taken early in your college career. Many states also require Core Academic Skills test scores as part of their teacher licensing process.

Individuals entering the teaching profession take the Praxis content and pedagogy tests as part of the teacher licensing and certification process required by many states. In addition, some professional associations and organizations require the Praxis Subject Assessments for professional licensing.

Do all states require these tests?
The Praxis tests are currently required for teacher licensure in approximately 40 states and United States territories. These tests are also used by several professional licensing agencies and by several hundred colleges and universities. Teacher candidates can test in one state and submit their scores in any other state that requires Praxis testing for licensure. You can find details at www.ets.org/praxis/states.

What is licensure/certification?
Licensure in any area—medicine, law, architecture, accounting, cosmetology—is an assurance to the public that the person holding the license possesses sufficient knowledge and skills to perform important occupational activities safely and effectively. In the case of teacher licensing, a license tells the public that the individual has met predefined competency standards for beginning teaching practice.

Because a license makes such a serious claim about its holder, licensure tests are usually quite demanding. In some fields, licensure tests have more than one part and last for more than one day. Candidates for licensure in all fields plan intensive study as part of their professional preparation. Some join study groups, others study alone. But preparing to take a licensure test is, in all cases, a professional activity. Because a licensure exam surveys a broad body of knowledge, preparing for a licensure exam takes planning, discipline, and sustained effort.

Why does my state require the Praxis tests?
Your state chose the Praxis tests because they assess the breadth and depth of content—called the “domain”—that your state wants its teachers to possess before they begin to teach. The level of content knowledge, reflected in the passing score, is based on recommendations of panels of teachers and teacher educators in
Appendix: Other Questions You May Have

each subject area. The state licensing agency and, in some states, the state legislature ratify the passing scores that have been recommended by panels of teachers.

How were the tests developed?
ETS consulted with practicing teachers and teacher educators around the country during every step of the Praxis test development process. First, ETS asked them what knowledge and skills a beginning teacher needs to be effective. Their responses were then ranked in order of importance and reviewed by hundreds of teachers.

After the results were analyzed and consensus was reached, guidelines, or specifications, for the selected-response and constructed-response tests were developed by teachers and teacher educators. Following these guidelines, teachers and professional test developers created test questions that met content requirements and ETS Standards for Quality and Fairness.*

When your state adopted the research-based Praxis tests, local panels of teachers and teacher educators evaluated each question for its relevance to beginning teachers in your state. During this “validity study,” the panel also provided a passing-score recommendation based on how many of the test questions a beginning teacher in your state would be able to answer correctly. Your state's licensing agency determined the final passing-score requirement.

ETS follows well-established industry procedures and standards designed to ensure that the tests measure what they are intended to measure. When you pass the Praxis tests your state requires, you are proving that you have the knowledge and skills you need to begin your teaching career.

How are the tests updated to ensure the content remains current?
Praxis tests are reviewed regularly. During the first phase of review, ETS conducts an analysis of relevant state and association standards and of the current test content. State licensure titles and the results of relevant job analyses are also considered. Revised test questions are then produced following the standard test development methodology. National advisory committees may also be convened to review and revise existing test specifications and to evaluate test forms for alignment with the specifications.

How long will it take to receive my scores?
Scores for tests that do not include constructed-response questions are available on screen immediately after the test. Scores for tests that contain constructed-response questions or essays aren't available immediately after the test because of the scoring process involved. Official score reports are available to you and your designated score recipients approximately two to three weeks after the test date for tests delivered continuously, or two to three weeks after the testing window closes for other tests. See the test dates and deadlines calendar at http://www.ets.org/praxis/register/dates_centers for exact score reporting dates.

Can I access my scores on the web?
All test takers can access their test scores via My Praxis Account free of charge for one year from the posting date. This online access replaces the mailing of a paper score report.

The process is easy—simply log into My Praxis Account at www.ets.org/praxis and click on your score report. If you do not already have a Praxis account, you must create one to view your scores.

Note: You must create a Praxis account to access your scores, even if you registered by mail or phone.

Your teaching career is worth preparing for, so start today!
Let the Praxis® Study Companion guide you.

To search for the Praxis test prep resources that meet your specific needs, visit:

www.ets.org/praxis/testprep

To purchase official test prep made by the creators of the Praxis tests, visit the ETS Store:

www.ets.org/praxis/store

Copyright © 2018 by Educational Testing Service. All rights reserved. ETS, the ETS logo, GRE, PRAXIS, and MEASURING THE POWER OF LEARNING are registered trademarks of Educational Testing Service (ETS). All other trademarks are property of their respective owners.

Measuring the Power of Learning.

www.ets.org