# Table of Contents

About the Assessment ............................................................................................................... 4  
Content Specifications ............................................................................................................. 5  
Test Subareas .......................................................................................................................... 6  
Test Objectives ....................................................................................................................... 6  
  Subarea I: Technology Applications Core ........................................................................... 6  
  Subarea II: Program Design and Development ................................................................. 8  
  Subarea III: Programming Language Topics .....................................................................10  
Code Segments .......................................................................................................................12  
  Example 1 ..........................................................................................................................12  
  Example 2 ..........................................................................................................................12  
  Example 3 ..........................................................................................................................13  
Practice Questions ....................................................................................................................14  
Answer Key ...............................................................................................................................31  
Preparation Resources .............................................................................................................32  
  Guide to Taking a GACE Computer-delivered Assessment ..................................................32  
  Reducing Test Anxiety ........................................................................................................32  
  Study Tips: Preparing for a GACE Assessment .................................................................32  
  Journals ...............................................................................................................................32  
  State-adopted Instructional Materials ................................................................................32  
  Other Resources ................................................................................................................32  
  Online Resources .............................................................................................................33  

**Note:** After clicking on a link, right click and select "Previous View" to go back to original text.
About the Assessment

<table>
<thead>
<tr>
<th>Assessment Name</th>
<th>Computer Science</th>
</tr>
</thead>
<tbody>
<tr>
<td>Grade Level</td>
<td>8–12</td>
</tr>
<tr>
<td>Test Code</td>
<td>554</td>
</tr>
<tr>
<td>Testing Time</td>
<td>3 hours</td>
</tr>
<tr>
<td>Test Duration</td>
<td>3.5 hours</td>
</tr>
<tr>
<td>Number of Selected-response Questions</td>
<td>100</td>
</tr>
<tr>
<td>Number of Constructed-response Questions</td>
<td>0</td>
</tr>
<tr>
<td>Test Format</td>
<td>Computer delivered</td>
</tr>
</tbody>
</table>

The GACE Computer Science assessment is designed to measure the professional knowledge of prospective teachers of computer science in the state of Georgia.

The testing time is the amount of time you will have to answer the questions on the test. Test duration includes time for tutorials and directional screens that may be included in the test.

The total number of questions that are scored is typically smaller than the total number of questions on the test. Most tests that contain selected-response questions also include embedded pretest questions, which are not used in calculating your score. By including pretest questions in the assessment, ETS is able to analyze actual test-taker performance on proposed new questions and determine whether they should be included in future versions of the test.
Content Specifications
This assessment is organized into content **subareas**. Each subarea is further defined by a set of **objectives** and their **knowledge statements**.

- The objectives broadly define what an entry-level educator in this field in Georgia public schools should know and be able to do.
- The knowledge statements describe in greater detail the knowledge and skills eligible for testing.
- Some tests also include content material at the evidence level. This content serves as descriptors of what each knowledge statement encompasses.

See a breakdown of the subareas and objectives for this assessment on the following pages.
Test Subareas

<table>
<thead>
<tr>
<th>Subarea</th>
<th>Approx. Percentage of Test</th>
</tr>
</thead>
<tbody>
<tr>
<td>I.   Technology Applications Core</td>
<td>33.3%</td>
</tr>
<tr>
<td>II.  Program Design and Development</td>
<td>33.3%</td>
</tr>
<tr>
<td>III. Programming Language Topics</td>
<td>33.3%</td>
</tr>
</tbody>
</table>

Test Objectives

Subarea I: Technology Applications Core

Objective 1: Knows technology terminology and concepts; the appropriate use of hardware, software, and digital files; and how to acquire, analyze, and evaluate digital information

The beginning Computer Science teacher:

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

Note: After clicking on a link, right click and select "Previous View" to go back to original text.
L. Demonstrates knowledge of intellectual property rights and related issues (e.g., copyright laws, fair use, patents, trademarks) when using, manipulating, and editing electronic data

M. Knows how to use online help and other support documentation

N. Knows how to use technical-writing strategies to develop documentation for a variety of communication products

O. Demonstrates knowledge of the impact of technology on society and the importance of technology to future careers, lifelong learning, and daily living for individuals of all ages

P. Investigates measures (e.g., passwords, virus detection/prevention) to protect computer systems and databases from unauthorized use and tampering

Objective 2: Knows how to use technology tools to solve problems, evaluate results, and communicate information in a variety of formats for diverse audiences

The beginning Computer Science teacher:

A. Knows how to plan, create, and edit documents using word processing features (e.g., readable fonts, alignment, page setup, tabs, ruler settings) to solve problems and communicate results

B. Knows how to plan, create, and edit spreadsheets using spreadsheet features (e.g., data types, formulas, functions, charts) to solve problems and communicate results

C. Knows how to plan, create, and edit databases using database features (e.g., defining fields, entering data, creating horizontal and vertical layouts) to solve problems and communicate results

D. Knows how to integrate one or more objects (e.g., tables, charts, graphs, graphics) into a product

E. Knows how to use productivity tools to create products (e.g., slide shows, posters, multimedia presentations, spreadsheets) for defined audiences

F. Knows how to publish information in a variety of ways (e.g., printed copy, monitor displays, Internet documents, video)

G. Knows how to use telecommunications tools (e.g., Internet browsers, video conferencing, distance learning) for a variety of purposes

H. Knows how to use interactive virtual environments (e.g., virtual field trips, instructional simulations)

I. Knows how to use collaborative software

J. Knows how to share information through online communication

K. Demonstrates knowledge of issues concerning proper etiquette when communicating using electronic tools

L. Demonstrates knowledge of how to design and implement procedures to track trends, set timelines, and review and evaluate products using technology tools (e.g., database managers, daily/monthly planners, project management tools)

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

Note: After clicking on a link, right click and select “Previous View” to go back to original text.
N. Knows how to select representative products to be collected and stored in an electronic evaluation tool and how to evaluate products for relevance to the assignment or task

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

Objective 3: \textit{Knows how to plan, organize, deliver, and evaluate instruction that effectively utilizes current technology for teaching technology applications for all students}

The beginning Computer Science teacher:

A. Knows how to plan computer science lessons using a range of instructional strategies for individuals and groups

B. Demonstrates knowledge of issues related to the equitable use of technology (e.g., gender, ethnicity, language, disabilities, access to technology)

C. Knows how to plan and implement instruction that allows students to use computer science in problem-solving and decision-making situations

D. Knows how to develop and facilitate collaborative tasks and teamwork among group members

E. Knows how to use technology tools to perform administrative tasks (e.g., attendance, grades, communication)

F. Knows how to use a variety of instructional strategies to ensure students’ reading comprehension

G. Knows strategies to help students learn how to locate, retrieve, analyze, evaluate, communicate, and retain content-related information

H. Knows how to evaluate student projects and portfolios using formal and informal assessment methods

I. Knows the relationship between instruction and assessment and uses assessment results for gauging student progress and adjusting instruction

J. Identifies resources to keep current with the use of technology in education and issues related to legal and ethical use of technology resources

K. Knows how to use technology to participate in self-directed activities in society and how to participate within electronic communities in a variety of roles (e.g., collaborator, learner, contributor, teacher/mentor)

\textbf{Subarea II: Program Design and Development}

Objective 1: \textit{Knows problem-solving strategies and different procedures for program design}

The beginning Computer Science teacher:

A. Exhibits knowledge of the analysis and design phases of the software system life cycle

B. Knows the characteristics of programming design strategies

C. Knows how to apply problem-solving strategies (e.g., design specification, top-down design, step-wise refinement, object-oriented design)
D. Demonstrates the ability to compare and contrast design strategies (e.g., top-down, bottom-up, object-oriented)

E. Demonstrates the use of visual organizers (e.g., flowcharts, schematic drawings) to design solutions to problems

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

Objective 2: Knows procedures for software development and implementation

The beginning Computer Science teacher:

A. Knows the characteristics of models (e.g., waterfall, incremental, spiral) used in the development of software systems

B. Knows how to survey the issues accompanying the development of large software systems (e.g., design/implementation teams, software validation/testing, risk assessment)

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

D. Knows how to create robust programs with emphasis on style, clarity of expression, and documentation to facilitate maintenance, program expansion, reliability, validity, and efficiency

E. Knows how to create and use libraries of generic modular code to be used for efficient programming

F. Demonstrates the ability to read and modify large programs, including design description and process development

G. Demonstrates effective use of predefined input and output, including logic to protect from invalid input

H. Demonstrates the ability to debug and solve problems using reference materials and effective strategies

I. Knows how to determine and employ methods to evaluate the design and functionality of information acquisition processes and algorithms, using effective coding, design, and test data

Objective 3: Knows computer science terminology and concepts and the characteristics of different programming languages and paradigms

The beginning Computer Science teacher:

A. Knows necessary vocabulary related to computer science (e.g., cache, bits, encryption)

B. Knows specific programming terminology (e.g., data type, data structure, encapsulation) and programming concepts (e.g., procedural, object-oriented)

C. Demonstrates knowledge of advanced computer science concepts (e.g., computer architecture, operating systems, artificial intelligence)
D. Demonstrates the ability to use notation for language definition (e.g., syntax diagrams, Backus-Naur forms)
E. Knows the differences in the levels of languages (e.g., machine, assembly, high-level compiled, interpreted)
F. Knows the characteristics of and differences in current programming languages and paradigms
G. Demonstrates knowledge of the uses of current programming languages and paradigms in other fields of study

Subarea III: Programming Language Topics

Objective 1: Correctly and efficiently uses data types, data structures, and functions in the development of code

The beginning Computer Science teacher:

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

Objective 2: Correctly and efficiently uses statements and control structures in the development of code

The beginning Computer Science teacher:

A. Applies standard operators (e.g., arithmetic, relational, logical, assignment, increment/decrement, input/output) and correct operator precedence
B. Identifies the characteristics of control structures
C. Uses conditional control structures (e.g., if, if . . . else statements)
D. Constructs iterative control structures (e.g., for and while statements, do loops)
E. Uses pretest (e.g., for, while) and posttest (e.g., do . . . while) loops

Note: After clicking on a link, right click and select “Previous View” to go back to original text.
F. Uses sequential, conditional, selection, and repetition execution control structures such as menu-driven programs that branch and allow user input

G. Demonstrates coding proficiency in contemporary programming languages, including an object-oriented language

Objective 3: Knows how to construct, compare and analyze various algorithms

The beginning Computer Science teacher:

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

B. Constructs sorting algorithms (e.g., selection, bubble, insertion, merge, shell, and quick sorts)

C. Compares and contrasts searching and sorting algorithms for space and time requirements

D. Constructs and appropriately uses iterative and recursive algorithms

E. Compares and contrasts iterative and recursive algorithms

F. Develops sequential, iterative, and recursive algorithms and codes programs in prevailing computer languages to solve practical problems

G. Analyzes various algorithms using “big-O” notation and best-, average-, and worst-case space and time techniques

H. Identifies and describes the correctness and complexity of specific types of algorithms (e.g., divide and conquer, greedy, backtracking)
## Code Segments

### Example 1

The following are some examples of pseudocode stimulus material.

Class declaration and object instantiation

```plaintext
class StudentInfo
    int studentID
    string name
end class StudentInfo

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

### Example 2

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

```plaintext
void f ( pass-by-reference int x, pass-by-value int y )
    x ← y + 1
    y ← x + 2
end f
```
Example 3

InsertionSort

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

The practice questions in this study companion are designed to familiarize you with the types of questions you may see on the assessment. While they illustrate some of the formats and types of questions you will see on the test, your performance on these sample questions should not be viewed as a predictor of your performance on the actual test. Fundamentally, the most important component in ensuring your success is familiarity with the content that is covered on the assessment.

To respond to a practice question, choose one of the answer options listed. Be sure to read the directions carefully to ensure that you know what is required for each question. You may find it helpful to time yourself to simulate actual testing conditions. A correct answer and a rationale for each sample test question are in the section following the practice questions.

Keep in mind that the test you take at an actual administration will have different questions, although the proportion of questions in each subarea will be approximately the same. You should not expect the percentage of questions you answer correctly in these practice questions to be exactly the same as when you take the test at an actual administration, since numerous factors affect a person's performance in any given testing situation.
Directions: Each of the questions or incomplete statements below is followed by four suggested answers or completions. Select the one that is best in each case.

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

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

   Answer

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

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

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

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

Answer

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

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

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

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

Which of the following formulas would NOT be a correct calculation of the six-week weighted average for Jane?

- A. \( =B2*0.15+C2*0.15+D2*0.15+E2*0.1+F2*0.1+G2*0.1+H2*0.1+I2*0.15 \)
- B. \( =(B2+C2+D2+I2)*0.15+(E2+F2+G2+H2)*0.1 \)
- C. \( =((B2+C2+D2+I2)*1.5+(E2+F2+G2+H2))/10 \)
- D. \( =(B2+C2+D2+I2)/15+(E2+F2+G2+H2)/10 \)

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

A. participate in a threaded discussion group.
B. send e-mail messages with attached document files.
C. update pages on the class’s website.
D. engage in dialogue in a real-time chat room.

Answer

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

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

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

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

A. \[
\begin{align*}
&\text{int part} \leftarrow 0 \\
&\text{int k} \leftarrow 1 \\
&\text{while } ( k < len ) \\
&\quad \text{if } ( \text{arr}[k] \leq \text{arr}[0] ) \\
&\quad\quad \text{part} \leftarrow \text{part} + 1 \\
&\quad\quad \text{swap } ( \text{arr}, \text{part}, k ) \\
&\quad \text{end if} \\
&\quad k \leftarrow k + 1 \\
&\text{end while}
\end{align*}
\]

B. \[
\begin{align*}
&\text{int part} \leftarrow 0 \\
&\text{int k} \leftarrow 1 \\
&\text{while } ( k < len ) \\
&\quad k \leftarrow k + 1 \\
&\quad \text{if } ( \text{arr}[k] \leq \text{arr}[0] ) \\
&\quad\quad \text{part} \leftarrow \text{part} + 1 \\
&\quad\quad \text{swap } ( \text{arr}, \text{part}, k ) \\
&\quad \text{end if} \\
&\text{end while}
\end{align*}
\]
C. \texttt{int part} \leftarrow 0 \\
\texttt{int k} \leftarrow 1 \\
\texttt{while ( k < len )} \\
\quad \texttt{if ( arr[k] > arr[0] )} \\
\qquad \texttt{part} \leftarrow \texttt{part + 1} \\
\qquad \texttt{swap ( arr, part, k )} \\
\quad \texttt{end if} \\
\quad \texttt{k} \leftarrow \texttt{k + 1} \\
\texttt{end while}

D. \texttt{int part} \leftarrow 0 \\
\texttt{int k} \leftarrow 1 \\
\texttt{while ( k < len )} \\
\quad \texttt{if ( arr[k] > arr[0] )} \\
\qquad \texttt{k} \leftarrow \texttt{k + 1} \\
\qquad \texttt{if ( arr[k] > arr[0] )} \\
\qquad \qquad \texttt{part} \leftarrow \texttt{part + 1} \\
\qquad \qquad \texttt{swap ( arr, part, k )} \\
\quad \texttt{end if} \\
\texttt{end while}

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

A. Incorporation of diagrams into the design documents  \\
B. Minimal use of global variables in the source code  \\
C. Construction of methods that are highly dependent on one another  \\
D. Extensive use of functions that support the debugging process

Answer

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

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

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

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

Answer

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

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

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

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

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

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

Answer

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

   A. ASCII
   B. Big-endian
   C. NUXI ordering
   D. Huffman encoding

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

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

   Answer

16. A binary heap data structure is best represented conceptually using which of the following?

   A. A binary tree
   B. A graph
   C. A linked list
   D. A stack

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

\[
\text{procedure calc ( pass-by-value int } w, \\
\text{pass-by-value int } x, \\
\text{pass-by-reference int } y, \\
\text{pass-by-reference int } z \}
\]

\[
w \leftarrow w + 1 \\
x \leftarrow x \times 2 \\
y \leftarrow y + 3 \\
z \leftarrow z \times 4
\]

\text{end procedure}

What are the values of \( a \) and \( b \) at the end of the code fragment below?

\[
\text{int } a \leftarrow 5 \\
\text{int } b \leftarrow 6 \\
calc ( a, a, b, b )
\]

A. \( a = 5 \) and \( b = 24 \)  
B. \( a = 5 \) and \( b = 36 \)  
C. \( a = 10 \) and \( b = 6 \)  
D. \( a = 12 \) and \( b = 6 \)

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

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

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

What is printed by the last line of code?

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

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

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

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

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

A. \( n > 0 \)
B. \( n ≥ 0 \)
C. \( n < 0 \)
D. \( n ≤ 0 \)

**Answer**
20. Consider the following pseudocode fragment, where \( x \) is an integer variable initialized to a nonnegative integer value.

```plaintext
// x is a nonnegative integer
int sum

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

Which of the following will calculate the same value of \( \text{sum} \) as the fragment above?

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

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

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

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

Answer

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

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

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

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

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

A. 2
B. 16
C. 16
   8
   4
   2
D. 2
   4
   8
   16

**Answer**

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

A. Quicksort
B. Insertion sort
C. Heapsort
D. Selection sort

**Answer**
24. Consider the following pseudocode binary search function, which returns the largest array index \( k \) such that \( a[k] \leq x \).

```plaintext
// precondition 1: integer array a is sorted in ascending order
// precondition 2: 0 \leq first < last < length of array a
// precondition 3: a[first] \leq x < a[last]
int f(int array a, int x, int first, int last)

while ( first + 1 \neq last )

    int mid ← ( first + last ) / 2      // integer division
    if ( x < a[mid] )

        last ← mid
    else
        first ← mid
    end if
end while

return first
end f
```

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

```plaintext
int f(int array a, int x, int first, int last)
if ( first + 1 == last )

    return first
end if

int mid ← ( first + last ) / 2
// missing code block
end f
```

Note: After clicking on a link, right click and select "Previous View" to go back to original text.
Which of the following could replace the missing code block so that the recursive function will work as intended?

A. \[
\text{if } ( x \geq a[mid] ) \quad \text{end if}
\]
\[
\text{return } f(a, x, first, mid)
\]
\[
\text{return } f(a, x, mid, last)
\]

B. \[
\text{if } ( x \geq a[mid] ) \quad \text{end if}
\]
\[
\text{return } f(a, x,_mid, first)
\]
\[
\text{return } f(a, x, last, mid)
\]

C. \[
\text{if } ( x \geq a[mid] ) \quad \text{end if}
\]
\[
\text{return } f(a, x, mid, last)
\]
\[
\text{return } f(a, x, first, mid)
\]

D. \[
\text{if } ( x \geq a[mid] ) \quad \text{end if}
\]
\[
\text{return } f(a, x, last, mid)
\]
\[
\text{return } f(a, x, mid, first)
\]

Answer

25. Consider the following pseudocode function.

// precondition: n and k are nonnegative integers
\[
\text{int } f ( \text{ int } n, \text{ int } k )
\]
\[
\text{if } ( k * n == 0 )
\]
\[
\text{return } 1
\]
\[
\text{else}
\]
\[
\text{return } f ( n - 1, k - 1 ) + f ( n - 1, k )
\]
\[
\text{end if}
\]
\[
\text{end } f
\]

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

A. 4
B. 5
C. 7
D. 11

Answer
<table>
<thead>
<tr>
<th>Question Number</th>
<th>Correct Answer</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>B</td>
</tr>
<tr>
<td>2</td>
<td>C</td>
</tr>
<tr>
<td>3</td>
<td>A</td>
</tr>
<tr>
<td>4</td>
<td>C</td>
</tr>
<tr>
<td>5</td>
<td>D</td>
</tr>
<tr>
<td>6</td>
<td>A</td>
</tr>
<tr>
<td>7</td>
<td>B</td>
</tr>
<tr>
<td>8</td>
<td>A</td>
</tr>
<tr>
<td>9</td>
<td>B</td>
</tr>
<tr>
<td>10</td>
<td>D</td>
</tr>
<tr>
<td>11</td>
<td>A</td>
</tr>
<tr>
<td>12</td>
<td>C</td>
</tr>
<tr>
<td>13</td>
<td>C</td>
</tr>
<tr>
<td>14</td>
<td>B</td>
</tr>
<tr>
<td>15</td>
<td>C</td>
</tr>
<tr>
<td>16</td>
<td>A</td>
</tr>
<tr>
<td>17</td>
<td>B</td>
</tr>
<tr>
<td>18</td>
<td>A</td>
</tr>
<tr>
<td>19</td>
<td>A</td>
</tr>
<tr>
<td>20</td>
<td>C</td>
</tr>
<tr>
<td>21</td>
<td>D</td>
</tr>
<tr>
<td>22</td>
<td>D</td>
</tr>
<tr>
<td>23</td>
<td>D</td>
</tr>
<tr>
<td>24</td>
<td>C</td>
</tr>
<tr>
<td>25</td>
<td>D</td>
</tr>
</tbody>
</table>

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

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

Guide to Taking a GACE Computer-delivered Assessment

This guide explains how to navigate through a GACE assessment and how to answer different types of test questions. This free download is available in the Test Preparation Resources section of the GACE website at www.gace.ets.org/prepare.

Reducing Test Anxiety

This guide provides practical help for people who suffer from test anxiety. Designed specifically for GACE test takers, but useful to anyone who has to take tests, this guide reviews the major causes of test anxiety and offers practical advice for how to counter each one.

Study Tips: Preparing for a GACE Assessment

This document contains useful information on preparing for selected-response and constructed-response tests. The instruction, tips, and suggestions can help you become a better-prepared test taker. See the Test Preparation Resources section of the GACE website at www.gace.ets.org/prepare for this free download.

Journals


Journal on Educational Resources in Computing, Association for Computing Machinery — jeric.acm.org


State-adopted Instructional Materials


Other Resources


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


Online Resources

AP Computer Science —
apcentral.collegeboard.com/apc/public/courses/teachers_corner/4483.html

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

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

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

Code.org — code.org/educate/curriculum

Code Highschool (codeHS) — codehs.com

CS Unplugged — csunplugged.org

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

Georgia Association of Educators (GAE) — pv.gae2.org

Georgia Computer Science Teachers Association (CSTA) — sites.google.com/site/georgiacsta

Georgia Department of Education — www.gapsc.com

Georgia Professional Standards Commission — www.gadoe.org

Note: After clicking on a link, right click and select "Previous View" to go back to original text.
Hour of Code — [hourofcode.com](http://hourofcode.com)
International Society for Technology in Education (ISTE) — [www.iste.org](http://www.iste.org)
Javabat (free online Java interactive learning tool) — [www.javabat.com](http://www.javabat.com)
National Center for Women & Information Technology (NCWIT) — [www.ncwit.org](http://www.ncwit.org)
Professional Association of Georgia Educators (PAGE) — [www.pageinc.org](http://www.pageinc.org)
Project Lead the Way (PLTW) — [www.pltw.org/our-programs/computer-science](http://www.pltw.org/our-programs/computer-science)
STEM Georgia — [www.stemgeorgia.org](http://www.stemgeorgia.org)

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