Florida Teacher Certification Examinations

Test Information Guide

for

Computer Science K–12

FLORIDA DEPARTMENT OF EDUCATION
www.fldoe.org

Fifth Edition

Developed, produced, and printed under the authority of the Florida Department of Education.

FTCE Administrator
Florida Department of Education
325 West Gaines Street, Suite 414
Tallahassee, Florida 32399-0400

Contents

1

Test and Test Information Guide Development

Teacher Certification Testing

Since 1980, Florida teacher certification candidates have been required to pass the Florida Teacher Certification Examinations (FTCE), which consisted of tests in reading, writing, mathematics, and professional knowledge. The 1986 Florida Legislature modified the testing program by also requiring teacher candidates to pass a test in the subject area in which they wish to be certified. In addition, the Legislature substituted the Florida College-Level Academic Skills Test (CLAST) for the reading, writing, and mathematics portions of the FTCE. The 2000 Florida Legislature replaced the CLAST with the General Knowledge Test, effective July 1, 2002.

The subject area knowledge tested on the Computer Science K–12 examination was identified and validated by committees of content specialists from within the state of Florida. Committee members included public school teachers, district supervisors, and college faculty with expertise in this field. Committee members were selected on the basis of recommendations by district superintendents, public school principals, deans of education, experts in the field, and other organizations. In developing the test, the committees used an extensive literature review, interviews with selected public school teachers, a large-scale survey of teachers, pilot tests, and their own professional judgment.

Role of the Test Information Guide

The purpose of this test information guide is to assist examinees in preparing for the subject area test in Computer Science K–12 and identifying areas in need of further study. The guide was designed to familiarize prospective test takers with various aspects of the examination, including the content that is covered and the way it is represented. The guide should enable candidates to direct their study and to focus on relevant material for review. An annotated bibliography of related subject matter materials is available at the end of the test information guide to provide further guidance in identifying additional areas in need of study.

This test information guide is intended primarily for use by certification candidates, who may be students in a college or university teacher-preparation program, teachers with provisional certification, teachers seeking certification in an additional subject area, or persons making a career change to public school teaching. Candidates may have studied and worked in Florida or may be from out of state.

College or university faculty may also use the guide to prepare students for certification, and inservice trainers may find the guide useful for helping previously certified teachers prepare for recertification or multiple certification.

This test information guide is not intended as an all-inclusive source of subject area knowledge, nor is it a substitute for college course work in the subject area. The sample questions are representative of the content of the actual test; however, they are not actual test questions from an actual test form. Instead, the guide is intended to help candidates prepare for the subject area test by presenting an overview of the content and format of the examination.

2 Preparation for the Test

The following outline may help you prepare for the examination. Adapt these suggestions to suit your own study habits and the time you have available for review.

Overview

- **Look over the organization of the test information guide.**

  Section 1 discusses the development of the test and test information guide.
  Section 2 (this section) outlines test preparation steps.
  Section 3 offers strategies for taking the test.
  Section 4 presents information about the content and structure of the test.
  Section 5 lists question formats and includes sample test questions.
  Section 6 provides an annotated bibliography of general references you may find useful in your review.
  Section 7 identifies a source of further information.

Self-Assessment

- **Decide which content areas you should review.**

  Section 4 includes the competencies and skills used to develop this subject area test and the approximate proportion of test questions from each competency area.

Review

- **Study according to your needs.**

  Review all of the competencies and concentrate on areas with which you are least familiar.

Practice

- **Acquaint yourself with the format of the examination.**

  Section 5 describes types of questions you may find on the examination.

- **Answer sample test questions.**

  Section 5 gives you an opportunity to test yourself with sample test questions and provides an answer key and information regarding the competency to which each question is linked.

Final preparation

- **Review test-taking advice.**

  Section 3 includes suggestions for improving your performance on the examination.

- **Refer to field-specific references.**

  Section 6 includes an annotated bibliography listing general references keyed to the competencies and skills used to develop this subject area test.

3 Test-Taking Advice

- Go into the examination prepared, alert, and well rested.
- Complete your travel arrangements prior to the examination date. Plan to arrive early so that you can locate the parking facilities and examination room without rushing.
- Dress comfortably and bring a sweater or jacket in case the room is too cool for your comfort.
- Take the following with you to the test site:
  - Admission ticket
  - Proper identification as described in "Identification Policy"
- There are many strategies for taking a test and different techniques for dealing with different types of questions. Nevertheless, you may find the following general suggestions useful.
  - Read each question and all the response options carefully before selecting your answer. Pay attention to all of the details.
  - Go through the entire test once and answer all the questions you are reasonably certain about. Then go back and work through the questions that require more thought.
  - When you are not certain of the correct answer, eliminate as many options as you can and choose the response that seems best. It is to your advantage to answer all the questions on the test, even if you are uncertain about some of your choices.
  - After completing the examination, go back and check every question. Verify that you have answered all of the questions and that your responses are correctly entered.

4 Competencies and Skills and Test Blueprint

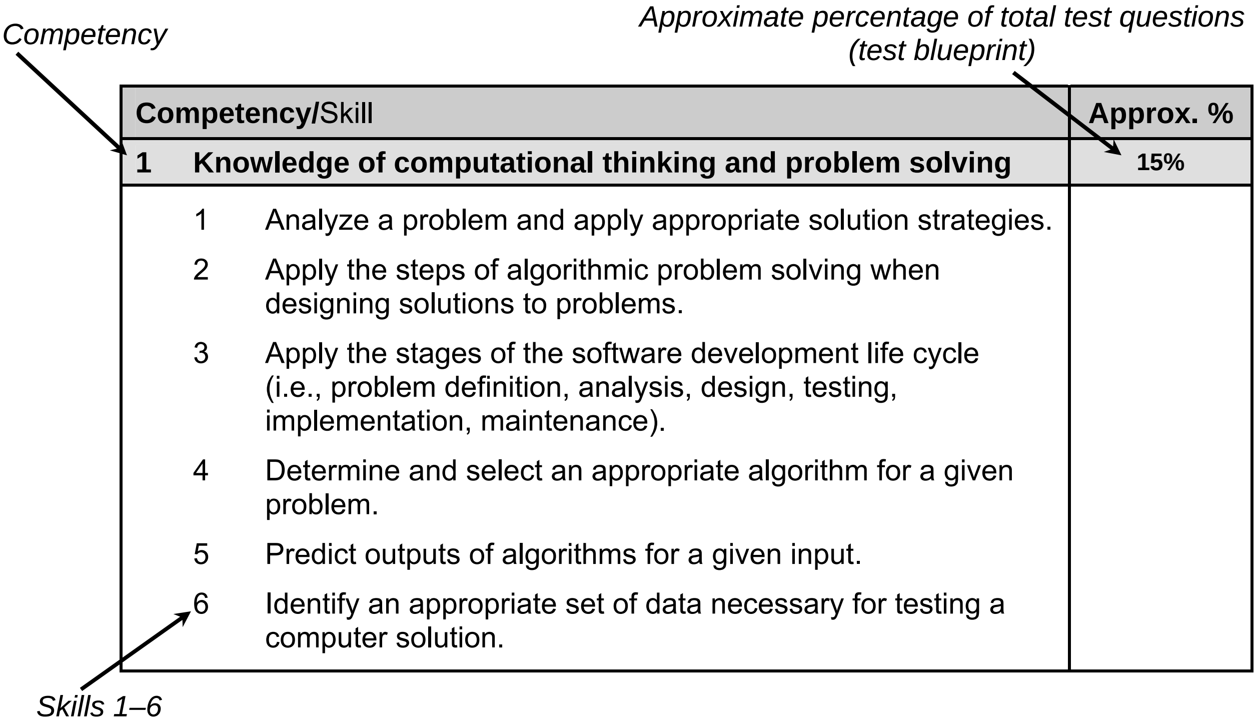
The table on the following pages lists the competencies and skills used as the basis for the Computer Science K–12 examination. These competencies and skills represent the knowledge that teams of teachers, subject area specialists, and district-level educators have determined to be important for beginning teachers. This table can serve as a checklist for assessing your familiarity with each of the areas covered by the test. The competencies and skills should help you organize your review. The test blueprint indicates the approximate percentage of test questions that will cover each specific competency on the exam.

Competencies are broad areas of content knowledge.

Skills identify specific behaviors that demonstrate the competencies.

Percentages indicate the approximate proportion of test questions that represent the competencies on the test.

The following excerpt illustrates the components of the table.


Competency
Approximate percentage of total test questions (test blueprint)
Skills 1–6

Competency/Skill			**Approx. %**
1	**Knowledge of computational thinking and problem solving**		**15%**
	1	Analyze a problem and apply appropriate solution strategies.	
	2	Apply the steps of algorithmic problem solving when designing solutions to problems.	
	3	Apply the stages of the software development life cycle (i.e., problem definition, analysis, design, testing, implementation, maintenance).	
	4	Determine and select an appropriate algorithm for a given problem.	
	5	Predict outputs of algorithms for a given input.	
	6	Identify an appropriate set of data necessary for testing a computer solution.	

Table of Competencies, Skills, and Approximate Percentages of Questions

Competency/Skill			**Approx. %**
1	**Knowledge of computational thinking and problem solving**		**15%**
	1	Analyze a problem and apply appropriate solution strategies.	
	2	Apply the steps of algorithmic problem solving when designing solutions to problems.	
	3	Apply the stages of the software development life cycle (i.e., problem definition, analysis, design, testing, implementation, maintenance).	
	4	Determine and select an appropriate algorithm for a given problem.	
	5	Predict outputs of algorithms for a given input.	
	6	Identify an appropriate set of data necessary for testing a computer solution.	
2	**Knowledge of data types and structures**		**15%**
	1	Distinguish between constants and variables and between local and global identifiers.	
	2	Distinguish between integer, real number, character, string, Boolean, and object data types.	
	3	Recognize and convert between binary, decimal, and hexadecimal number systems.	
	4	Identify characteristics and uses of data structures, including arrays, linked lists, stacks, queues, and sets.	
	5	Distinguish between instance, class, and local variables in an object-oriented design.	
	6	Identify components of class declarations for an object-oriented program and distinguish between public and private access specifiers.	

Competency/Skill	**Approx. %**
3 Knowledge of programming logic	**20%**
1 Distinguish between error types (e.g., syntax, runtime, logic) and apply principles of debugging.	
2 Identify principles, characteristics, and uses of internal and external program documentation.	
3 Analyze the characteristics and functions of object-oriented and procedural languages.	
4 Select the appropriate algorithmic sequence, conditional, iteration, and recursive constructs for a given purpose.	
5 Analyze characteristics and applications of searching (i.e., sequential, binary) and sorting (i.e., selection, insertion, merge) algorithms.	
6 Analyze the characteristics and applications of propositional logic (e.g., De Morgan's laws).	
4 Knowledge of programming languages	**20%**
1 Identify characteristics and apply concepts of the Scratch™[1] programming language learning environment from the MIT Media Library.	
2 Analyze segments of Java®[2] code containing sequential, conditional, or iteration statements.	
3 Analyze segments of Java® code involving methods, interacting objects, or passing parameters.	
4 Apply principles of data types and data manipulation (e.g., string methods, arithmetic operations) in the Java® programming language.	
5 Apply principles of abstraction, encapsulation, inheritance, and polymorphism in the Java® programming language.	

[1] The Scratch trademark is the property of MIT.

[2] Java is a registered trademark of Oracle and/or its affiliates.

Competency/Skill		Approx. %
5	**Knowledge of computer hardware, software, and networking**	**10%**
1	Identify the hardware components of a computer system and their functions (e.g., input, output, processing, storage).	
2	Analyze the advantages, disadvantages, or both of various data storage technologies.	
3	Identify the characteristics and uses of various types of software (e.g., system, application).	
4	Apply features and functions of application and productivity software (e.g., word processing, spreadsheet, database, multimedia authoring, Web development software).	
5	Identify concepts and terminology related to networks (e.g., network protocols, Open Systems Interconnection model, client-server, cloud computing).	
6	Identify characteristics and uses of network devices (e.g., servers, routers, switches, access points, workstations).	
6	**Knowledge of the historical aspects and social issues related to computer technologies**	**10%**
1	Identify examples of appropriate use (e.g., software license types, archival copying, fair use of copyrighted materials) and misuse (e.g., plagiarism, music and video piracy) of intellectual property.	
2	Identify milestones in the historical development of computer technology and important contributions of individuals or groups to the development of computer technology.	
3	Analyze cultural, legal, and ethical issues and responsibilities of digital citizens, organizations, and government entities (e.g., privacy issues related to Internet use, data protection).	
4	Analyze issues related to malicious software, social engineering, and security awareness.	
5	Identify concepts and terminology related to security countermeasures (e.g., firewalls, antivirus programs, filtering software, encryption) that prevent, detect, and correct breaches.	
6	Analyze security issues related to maintaining the confidentiality, integrity, and availability of information.	

Competency/Skill		**Approx. %**
7	**Knowledge of computer science pedagogy**	**10%**
1	Apply appropriate and effective classroom management strategies for teaching computer science (e.g., laboratory work, cooperative learning, electronic communications).	
2	Apply appropriate and effective instructional strategies for teaching computer science (e.g., independent learning, case studies, role-playing, manipulatives, visualizations, simulations, modeling, team software development).	
3	Apply appropriate and effective formative and summative assessment strategies for teaching computer science (e.g., rubrics, portfolios).	
4	Apply appropriate and effective accommodations, adaptations, and strategies that ensure the equitable use of technology for diverse student populations (e.g., students with exceptionalities, English language learners, students from various socioeconomic levels).	
5	Determine characteristics and apply uses of instructional technologies (e.g., collaborative online tools, social networking, computer-based learning, mobile devices).	
6	Recognize opportunities, skills, and paths related to college and career readiness in the field of computer science.	
7	Apply practices for planning and developing curricula that meet state and national standards and recognize resources for ongoing professional support and development.	

5 Test Format and Sample Questions

The Computer Science K–12 subject area test consists of approximately 120 multiple-choice questions. You will have two and one-half hours to complete the test.

Each question will contain four response options, and you will indicate your answer by selecting **A**, **B**, **C**, or **D**.

The table below presents types of questions on the examination and refers you to a sample question of each type.

Type of Question	Sample Question
Sentence Completion Select the response option that best completes the sentence.	Question 1, page 10
Scenario Examine a situation, problem, or case study. Then answer a question, make a diagnosis, or recommend a course of action by selecting the best response option.	Question 2, page 10
Command Select the best response option.	Question 4, page 11
Direct Question Choose the response option that best answers the question.	Question 6, page 12

Sample Questions

The following questions represent both the form and content of questions on the examination. These questions will acquaint you with the general format of the examination; however, these sample questions do not cover all of the competencies and skills that are tested and will only approximate the degree of examination difficulty.

An answer key follows at the end of the sample questions. The answer key includes information regarding the competency to which each question is linked.

DIRECTIONS: Read each question and select the best response.

1. Using string methods would be an appropriate part of a solution strategy for

A. reversing the letters in a word.
B. creating a multidimensional array.
C. printing a line of text to the screen.
D. constructing an instance of a class.

2. A programmer would like to determine the most efficient algorithm for ordering a large, finite list of integers from largest to smallest. The programmer has the following information on several sorting algorithms.

Algorithm	Average complexity
selection sort	n^2
merge sort	$n \log n$
bogosort	$n\, n!$
shell sort	$n (\log n)^2$

Which algorithm is likely to be most appropriate for this task?

A. selection sort
B. merge sort
C. bogosort
D. shell sort

3. The pseudocode below prompts the user to enter data from the keyboard one number at a time. Given the keyboard inputs 2, 5, 4, 5, and 4 in that order, what is the correct output of the following algorithm?

```
sum = 0
i = 0
while (i < 5) do {
 print "Input a number, n"
 input n
 if (n mod 2 is equal to 0) then sum = sum + n
 i = i + 1
} endwhile
avg_total = sum / i
output avg_total
```

A. 2
B. 4
C. 10
D. 20

4. Given the following program segment, select the minimum set of data required to test all cases.

```
print "What was your score on the test?"
input score
if (score ≥ 90)
   then print "Excellent!"
else if (score ≥ 75)
   then print "Not too bad."
else
   print "Study harder next time!"
```

A. 90, 70, 65
B. 90, 80, 70
C. 95, 70, 60
D. 95, 90, 75

5. Which variable in the following pseudocode program segment would be considered global?

```
function1(int w) {
    int x = 12
    /* some code */
}
int y = 5
mainProgram( ) {
    int z = 7
    function1(z)
    /* some more code */
}
```

A. w
B. x
C. y
D. z

6. A variable of type Boolean would be most appropriate to store the result of which of the following types of tasks?

A. object creation
B. concatenation
C. arithmetic calculation
D. object creation

7. Which of the following base ten numbers is equal to the hexadecimal number 3A9C?

A. $3 \times 6^3 + 10 \times 6^2 + 9 \times 6^1 + 12 \times 6^0$
B. $12 \times 6^3 + 9 \times 6^2 + 10 \times 6^1 + 3 \times 6^0$
C. $3 \times 16^3 + 10 \times 16^2 + 9 \times 16^1 + 12 \times 16^0$
D. $12 \times 16^3 + 9 \times 16^2 + 10 \times 16^1 + 3 \times 16^0$

8. Which of the following data structures would be most appropriate for a computer program to keep track of and choose from a list of applicants for available season football tickets?

   A. stack
   B. queue
   C. hash table
   D. binary tree

9. According to the pseudocode below, which line of the code constructs an instance of a class?

```
1)    import from library Rectangle;

2)    class TestRectangle{
3)           mainProgram() {
4)           int xcorner, ycorner, width, height
5)           xcorner = 50
6)           ycorner = 30
7)           width = 5
8)           height = 10

9)           Rectangle rect = new Rectangle(
                   xcorner, ycorner, width, height)
10)          print ("The x-coordinate of the rectangle is"
                   + rect.getX())
11)          }
12)   }
```

   A. 1
   B. 4
   C. 9
   D. 10

10. According to the code segment below, which method can only be accessed from within the class `Rectangle`?

```
public class Rectangle {
        private int mywidth
        private int myheight
        private int myarea
        private int myperim

        /* some code */

        private findArea( ) {
                myarea = mywidth* myheight
        }

        public findPerimeter( ) {
                myperim = mywidth + myheight
        }

        public setHeight(int height) {
                myheight = height
        }

        public setWidth(int width) {
                mywidth = width
        }
}
```

A. `findArea()`
B. `findPerimeter()`
C. `setHeight()`
D. `setWidth()`

11. The pseudocode below, used to determine if the user input is in the requested range, contains an error.

```
print "Type a number from 1 to 100, inclusive"
input number
if (number > 1) or (number < 100)
    then print "The number is in the correct range"
```

Which code segment corrects the error?

A. `if (number ≥ 1) and (number ≤ 100)`
B. `if (number ≥ 1) or (number ≤ 100)`
C. `if (number > 1) and (number < 100)`
D. `if (number < 1) or (number > 100)`

12. Which of the following best describes the primary reason for producing internal program documentation?

A. increasing program reliability by reporting and storing error codes
B. reducing execution time by improving memory allocation and usage
C. providing explanations of code segments for maintenance and updates
D. tracing the execution by printing output at critical branching points

13. In an object-oriented programming language, encapsulation is most likely to be implemented through the use of

A. private data and methods.
B. recursion in place of iteration.
C. subroutines in a modular structure.
D. functions containing nested loops.

14. When n = 5, which of the following loops will generate the output values 1, 2, 3, 4, 5, and 6?

A.
```
count = 1
while (count ≤ n) do {
     print count
     count = count + 1
} endwhile
```

B.
```
count = 1
while (count < n) do {
     count = count + 1
     print count

} endwhile
```

C.
```
count = 0
while (count < n) do {
     print count
     count = count + 1
} endwhile
```

D.
```
count = 0
while (count ≤ n) do {
     count = count + 1
     print count
} endwhile
```

15. A student has a horizontal list of numbered cards and sorts them manually by moving the cards in the sequence below.

    1. Find the smallest card in the list and exchange it with the first card in the list.
    2. Find the next smallest card in the list and exchange it with the second card in the list.
    3. Repeat these instructions until the original list is sorted.

    Which of the following algorithms is described above?

    A. bubble sort
    B. selection sort
    C. insertion sort
    D. merge sort

16. The following pseudocode is used to determine admission prices.

```
if ((not age > 55) and (not age < 22) and (haveCoupon =
False)) {
         admission = 15.00
}
else {
         admission = 10.00
}
```

Which of the following segments of pseudocode is equivalent?

A.
```
if ((age > 55) or (age < 22) or (haveCoupon = True))
           then admission = 10.00
else admission = 15.00
```

B.
```
if ((age > 55) and (age < 22) or (haveCoupon = True))
           then admission = 15.00
else admission = 10.00
```

C.
```
if ((age > 55) and (age < 22) and (haveCoupon = True))
           then admission = 10.00
else admission = 15.00
```

D.
```
if ((age > 55) or (age < 22) and (haveCoupon = True))
           then admission = 15.00
else admission = 10.00
```

17. A student creating a program using the Scratch™ programming environment from the MIT Media Library would most likely be engaged in which of the following activities?

    A. writing computer code to move a turtle around the computer screen
    B. snapping program blocks together to manipulate the properties of a sprite
    C. controlling a robot in an environment involving intersecting streets
    D. dragging and dropping graphic tiles to manipulate three-dimensional figures

18. What will be the output when the following section of Java® code runs?

```
int x = 5;
int y = 10;
int z;
z = x;
x = y;
y = z;
System.out.println(x + " " + y);
```

    A. 5 5
    B. 10 10
    C. 10 5
    D. 5 10

19. A Java® class is defined as follows.

```
public class TestingFunctions
{
        public void fun1(int a)
        {
                a++;
                System.out.println(a);
        }
        public void fun2(int b)
        {
                b--;
                this.fun1(b);
        }
}
```

In addition, the following lines of code are executed from the instance of another class that has access to the `TestingFunctions` class through an import statement.

```
int x = 5;
TestingFunctions test = new TestingFunctions( );
test.fun2(x);
```

What will be the output from these three lines of code?

A. 4
B. 5
C. 6
D. 7

20. According to the following segment of Java® code, which Java® string method should be used to print "Extra" to the screen?

```
String text = "Extra Special!";
String part = /* string method */
System.out.println(part)
```

A. `text.substring(0, 4)`
B. `text.substring(0, 5)`
C. `text.substring(1, 5)`
D. `text.substring(1, 6)`

21. Two Java® classes are defined as follows.

```
public class Class1 {
        public void m1( ) {
             System.out.print("one");
        }
        public void m2( ) {
             System.out.print("two");
             this.m1( );
        }
}
public class Class2 extends Class1 {
        public void m1( ) {
             System.out.print("three");
             super.m2( );
        }
        public void m2( ) {
             System.out.print("four");
        }
}
```

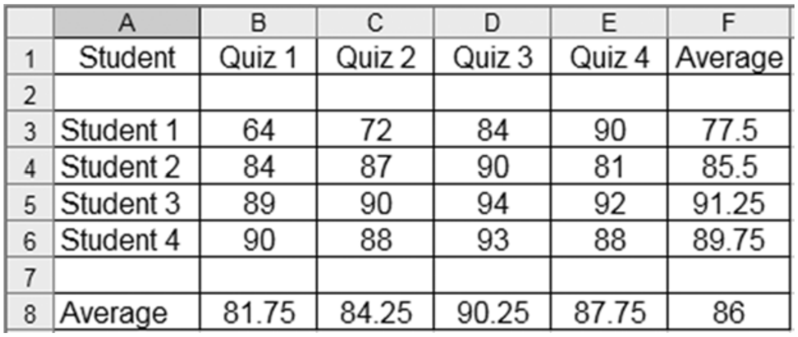
In addition, the following code appears in the instance of another class that has access to these two Java® classes by import statements.

```
Class2 number = new Class2 ( );
number.m1( );
```

What will be printed by these two lines of code?

A. one two three
B. one three two
C. three two one
D. three two three

22. A teacher has created the spreadsheet below to calculate average quiz grades for several students.



	A	B	C	D	E	F
1	Student	Quiz 1	Quiz 2	Quiz 3	Quiz 4	Average
2						
3	Student 1	64	72	84	90	77.5
4	Student 2	84	87	90	81	85.5
5	Student 3	89	90	94	92	91.25
6	Student 4	90	88	93	88	89.75
7						
8	Average	81.75	84.25	90.25	87.75	86

Which of the following formulas could be used to calculate the average quiz grade for Student 3?

A. =(B5:B8)/4
B. =(B3:B6)/4
C. =(B5:E5)/4
D. =(B6:E6)/4

23. At which of the following layers of the Open Systems Interconnection model is wireless network technology implemented?

A. application
B. session
C. data link
D. physical

24. Which of the following devices would be most appropriate for connecting a local area network using one protocol with a wide area network using a different protocol?

A. hub
B. server
C. router
D. switch

25. Which of the following best describes the malicious nature of worm software?

    A. giving a remote hacker unauthorized access to a network computer
    B. sending data from a host computer to another unauthorized entity
    C. self-replicating in order to spread across networks to other computers
    D. deleting or altering essential files stored on an infected computer

26. The most significant drawback associated with software that uses signature-based detection methods for identifying computer malware is that the software will

    A. delete essential files that are not infected.
    B. fail to identify new or unknown viruses.
    C. block incoming data from the network server.
    D. require frequent scans of all executable files.

27. A plan for ensuring the fault tolerance of a computer network would most likely contain a strategy for

    A. encrypting the most sensitive data.
    B. installing a proxy server on the host.
    C. maintaining uninterrupted power.
    D. increasing the network bandwidth.

28. A teacher is developing a rubric for assessing students' computer programs. One of the teacher's goals is that students will be able to explain how their programs work. Which of the following rubric components is most closely aligned with this goal?

    A. The program compiles without any syntax or runtime errors.
    B. The program outputs the correct solution to the problem.
    C. The code variables are declared using the correct data types.
    D. The code contains clear and appropriate internal documentation.

29. Which of the following types of computer software would be most effective in helping an English language learner make connections between written and spoken English?

A. word prediction
B. sound recording
C. speech synthesizer
D. automated translation

30. An elementary school teacher would like to pursue professional development opportunities for integrating computational thinking and programming concepts into the classroom. Which of the following organizations could provide the teacher with the most relevant resources to achieve this goal?

A. Computer Science Teachers Association
B. Special Interest Group on Computer Science Education
C. International Society for Technology in Education
D. Association for the Advancement of Computing in Education

Answer Key

Question Number	Correct Response	Competency
1.	A	1
2.	B	1
3.	A	1
4.	B	1
5.	C	2
6.	D	2
7.	C	2
8.	B	2
9.	C	2
10.	A	2
11.	A	3
12.	C	3
13.	A	3
14.	D	3
15.	B	3
16.	A	3
17.	B	4
18.	C	4
19.	B	4
20.	B	4
21.	C	4
22.	C	5
23.	D	5
24.	C	5
25.	C	6
26.	B	6
27.	C	6
28.	D	7
29.	C	7
30.	A	7

6 Annotated Bibliography

The annotated bibliography in this section includes basic references that you may find useful in preparing for the exam. Each resource is linked to the competencies and skills found in Section 4 of this guide.

This bibliography is representative of the most important and most comprehensive texts pertaining to the competencies and skills for Computer Science K–12. The Florida Department of Education does not endorse these references as the only appropriate sources for review; many comparable texts currently used in teacher preparation programs also cover the competencies and skills that are tested on the examination.

1. Brookshear, J.G., Smith, D.T., & Brylow, B. (2012). *Computer science: An overview* (11th ed.). Boston, MA: Addison-Wesley.

Focuses on the teaching of computer science concepts and processes for a broad audience of students who may or may not have previous experience in computer science. Useful for review of competency 1.

2. Dean, T. (2013). *Network+ guide to networks* (6th ed.). Boston, MA: Course Technology/Cengage Learning.

Specializes in teaching fundamental, vendor-independent networking concepts including the installation, configuration, and troubleshooting of networks for students with little or no prior experience in networking. Includes the Open Systems Interconnection model. Useful for review of competency 5.

3. Farrell, J. (2013). *Programming logic and design: Comprehensive version* (7th ed.). Boston, MA: Course Technology/Cengage Learning.

Addresses topics regarding programming logic, as well as the design and setup of programs, including coding etiquette and other good programming practices for computer science students. Useful for review of competencies 1 and 3.

4. Gray, S. (2007). *Data structures in Java: From abstract data types to the Java collections framework.* London: Addison-Wesley.

Deals primarily with the use of the Java Collections Framework in teaching students to develop programs manually using data structures and abstract data types. Useful for review of competencies 2 and 4.

5. Hazzan, O., Lapidot, T., & Ragonis, N. (2011). *Guide to teaching computer science: An activity-based approach.* London: Springer.

Focuses on the pedagogical and practical aspects of the teaching of computer science, using several targeted activities designed to teach students how to think logically and solve problems effectively. Useful for review of competencies 1 and 7.

6. Hoover, J.J., Klinger, J.K., Baca, L.M., & Patton, J.M. (2008). *Methods for teaching culturally and linguistically diverse exceptional learners.* Upper Saddle River, NJ: Pearson Prentice Hall.

Provides a wealth of information regarding educational best practices for teaching students who are culturally and/or linguistically diverse, including those who require differentiated instruction as part of their education. Useful for review of competency 7.

7. Horstmann, C. (2013). *Big Java: Late objects*. Hoboken, NJ: John Wiley & Sons, Inc.

Introduction to programming, and especially programming in the Java language, that focuses on fundamental skills and processes, as well as on effective student learning and classroom practice. Object-oriented concepts are introduced in later chapters. Useful for review of competencies 1, 2, 3, 4, and 7.

8. Litvin, M., & Litvin, G. (2006). *Java methods A & AB: Object-oriented programming and data structures*. Andover, MA: Skylight Publishing.

Specializes in object-oriented programming in the Java programming language, providing a thorough introduction to concepts, data structures, algorithms, and their implementations. Useful for review of competencies 2, 3, and 4.

9. Preston, J., Preston, S., & Ferrett, R.L. (2010). *Computer literacy for IC^3™ Unit 2: Using productivity software*. Upper Saddle River, NJ: Pearson Prentice Hall.

Introductory text for the teaching of computer and internet literacy, focused on the use of productivity software such as e-mail clients, word processors, databases, and others. Useful for review of competencies 5, 6, and 7.

10. Savitch, W., & Carrano, F. M. (2008). *Java^{TM}: An introduction to problem solving and programming* (5th ed.). Upper Saddle River, NJ: Pearson Prentice Hall.

Designed for teaching the Java programming language, computer science ideas, and programming techniques to a beginning audience with no prior programming experience. Useful for review of competencies 1, 2, 3, and 4.

11. SCRATCH Software (Version 1.4) [Computer software]. Cambridge, MA: Lifelong Kindergarten Group. Retrieved from http://scratch.mit.edu

Introductory programming interface created by the Massachusetts Institute of Technology for the teaching, learning, and practice of programming logic, coding, and other processes common to computer science in general and computer programming in specific. Useful for review of competency 4.

12. Smith, R.E. (2013). *Elementary information security*. Burlington, MA: Jones & Bartlett Learning.

Introductory text focusing exclusively on the technical, social, and process-oriented aspects of the field of information security. Useful for review of competencies 5 and 6.

7 Additional Information

Please visit the following website to review FTCE registration details and to find additional FTCE information, including test locations and passing scores.

www.fldoe.org/accountability/assessments/postsecondary-assessment/ftce/