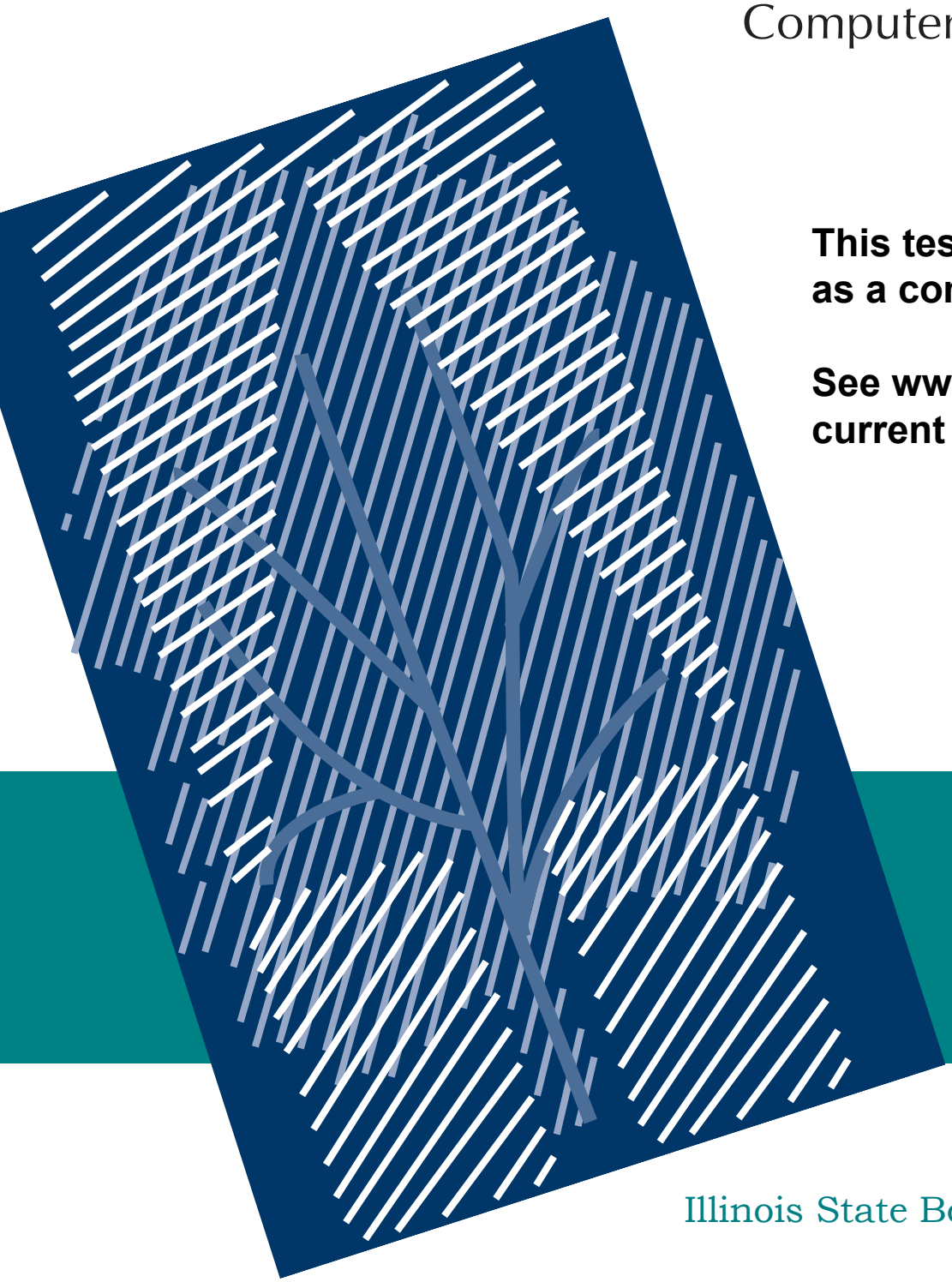
Illinois Licensure Testing System

STUDY GUIDE

Computer Science (038)



This test is now delivered as a computer-based test.

See www.il.nesinc.com for current program information.

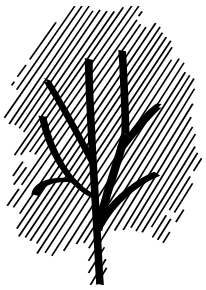
Illinois State Board of Education

IL-SG-FLD038-04

An Equal Opportunity/Affirmative Action Employer
Printed by the Authority of the State of Illinois

TABLE OF CONTENTS

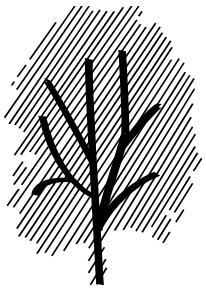
Readers should be advised that this study guide, including many of the excerpts used herein, is protected by federal copyright law.



General Information About the Illinois Licensure Testing System

The first section of the study guide is available in a separate PDF file. Click the link below to view or print this section.

General Information About the Illinois Licensure Testing System



Field-Specific Information

- **Test Subareas and Objectives**
- **Practice Test Questions**
- **Explanation of the Test Score Report**

INTRODUCTION

The content tests are designed to assess a candidate's knowledge of content in the specific teaching, school service personnel, or administrative field in which licensure is sought. The tests are based on current and relevant expectations for teacher preparation students and for teachers in Illinois as defined by the Illinois Content Area Standards for Educators. This study guide is designed to focus your preparation by helping you become familiar with the format and content to be covered on the tests.

This section includes a list of the test subareas and objectives, practice test questions for the field covered by this study guide, an answer key, and an explanation of the test score report.

TEST SUBAREAS AND OBJECTIVES

The content covered by the test is organized into subareas. You will find a list of subareas at the beginning of the list of test objectives. Within each subarea, the content is further defined by a set of objectives. Each objective comprises two major parts:

1. the *objective statement*, which broadly defines the knowledge and skills that an entry-level educator needs to know; and

2. the *descriptive statement*, which describes in greater detail the types of knowledge and skills covered by the test objective.

The test objectives are broad, conceptual, and meaningful statements, written in language that reflects the skills, knowledge, and understanding that an entry-level teacher needs in order to teach effectively in an Illinois classroom. A test consists of test questions that measure an examinee's mastery of these test objectives.

Below is an example of a test objective statement and its accompanying descriptive statement for the Computer Science test.

Objective Statement

Identify the basic features of computer systems.

Descriptive Statement

Includes the basic functions and components of computer systems, the basic functions and characteristics of stand-alone microcomputer systems, and criteria for evaluating computer systems.

PRACTICE TEST QUESTIONS

The practice test questions included in this section are designed to give the examinee an introduction to the nature of the test questions included on the ILTS test for each field. The practice test questions, which begin on page 2-8, represent the various types of test questions you may expect to see on an actual test; however, they are **not** designed to provide diagnostic information to help you identify specific areas of individual strengths and weaknesses or predict your performance on the test as a whole. Use the answer key located after the practice test questions to check your answers.

To help you identify which test objective is being assessed, the objective statement to which the question corresponds is listed in the answer key. When you are finished with the practice test questions, you may wish to go back and review the entire list of test objectives and descriptive statements once again.

COMPUTER SCIENCE TEST OBJECTIVES

I. Computer Systems
II. Programming Theory and Methods
III. Computer Languages, Data Concepts, and Algorithms
IV. Computer Applications and Assessment

SUBAREA I—COMPUTER SYSTEMS

1. Identify the basic features of computer systems.

Includes the basic functions and components of computer systems, the basic functions and characteristics of stand-alone microcomputer systems, and criteria for evaluating computer systems.

2. Identify the general architectural characteristics of computers.

Includes the components of processing devices and their functions; and types and characteristics of storage devices.

3. Identify types and characteristics of peripheral devices.

Includes types of input devices and their characteristics (e.g., joystick, light pen, graphics tablet, keyboard, optical scanner, mouse, voice recognition unit); types of output devices and their characteristics; terms related to input and output devices; and types and characteristics of telecommunications devices.

4. Understand the general principles of network connectivity.

Includes the historical development of computer networks, characteristics of a computer network (e.g., communication protocols, topologies), network components (e.g., file servers, bridges), and the Internet.

5. Identify types of programming software and software applications.

Includes types of software packages and their applications; software appropriate for various situations; and the functions of compilers, interpreters, assemblers, linkers, editors, loaders, and macros.

6. Identify criteria for evaluating software.

Includes criteria for evaluating educational software, applications software, and telecommunications software.

7. Understand the functions and characteristics of operating systems.

Includes functions of an operating system; and types and characteristics of operating systems utilities.

SUBAREA II—PROGRAMMING THEORY AND METHODS

8. Understand the function of flowcharts and pseudocode in program preparation.

Includes the functions of flowcharts and pseudocode in program preparation; and the interpretation of computer flowcharts and pseudocode.

9. Apply modularization in program design.

Includes the function of the module in program design, methods for decomposing programs into modules, methods for decomposing modules into submodules, and modularization strategies in programming situations.

10. Analyze design strategies in program development.

Includes the steps in a programming process, the purpose and function of design strategies, differences between top-down and bottom-up designs, and the advantages of different design strategies.

11. Apply the principles of coding in program development.

Includes the structure of program code; and techniques for developing code.

12. Understand the principles of testing computer programs.

Includes the function of program testing, strategies for designing test plans including interactive and noninteractive programs, the sequence of steps in running a program test, and criteria for evaluating test plans.

13. Apply debugging procedures.

Includes common programming errors, procedures for locating program errors, strategies for debugging programs, and program errors in program listings and printed output.

14. Identify principles of program documentation.

Includes reasons for documenting computer programs, types and characteristics of program documents, and types and characteristics of internal and external documentation.

SUBAREA III—COMPUTER LANGUAGES, DATA CONCEPTS, AND ALGORITHMS

15. Analyze declarations and data types common to high-level languages using a pseudolanguage.

Includes the function and characteristics of variable declarations, data types, and identifiers; and the interpretation of code for correct use of global and local identifiers.

16. Understand constants and variables common to high-level languages using a pseudolanguage.

Includes types and characteristics of constants and variables; and the interpretation of code for appropriate use of data as represented by constants and variables.

17. Understand statements and operators common to high-level languages using a pseudolanguage.

Includes purposes of statements and operators, types and characteristics of operators, the purpose and function of operator precedence, and the purpose and function of assignment statements.

18. Understand control structures common to high-level languages using a pseudolanguage.

Includes characteristics of sequential execution structures; characteristics of conditional execution structures with or without branching; characteristics of repetitive execution structures; and the interpretation of code for correct use of sequential, conditional, or repetitive execution control structures.

19. Identify input and output procedures common to high-level languages using a pseudolanguage.

Includes the function and characteristics of terminal input and output procedures and of file input and output procedures.

20. Understand subprograms, procedures, and functions common to high-level languages using a pseudolanguage.

Includes the purpose and function of parameters; and the interpretation of code for the correct use of parameter passing techniques.

21. Apply methods of program annotation common to high-level languages using a pseudolanguage.

Includes the role and function of comments, formatting, and indentation; and the application of program annotation in various situations.

22. Identify and analyze features and uses of common programming languages.

Includes programming features and common applications of BASIC, LOGO, PASCAL, and C.

23. Understand the internal representation of data.

Includes types and functions of number systems, the application of number systems skills, and characteristics of codes.

24. Identify the functions and uses of data structures.

Includes the functions of data structures, such as arrays, strings, linked lists, stacks, and queues; and the uses of data structures.

25. Understand representation of data structures.

Includes procedures for representation of data structures; and characteristics of representation of data structures.

26. Use algorithms to manipulate data.

Includes techniques for manipulating data structures using string processing, algorithmic operations commonly performed on data structures, and the application of procedures for manipulating data structures.

27. Identify characteristics and functions of algorithms.

Includes characteristics and functions of searching algorithms and sorting algorithms.

SUBAREA IV—COMPUTER APPLICATIONS AND ASSESSMENT

28. Understand the historical development of computers.

Includes major persons and events in the development of computers; the generations of computer development and their characteristics; differences between analog and digital computers; and significant technological aspects of the development of personal computers.

29. Analyze the impact of computers on society.

Includes the growth of the computer industry, the impact of computers on communication and transportation, the impact of computers on entertainment and work life, and the impact of computers on the job market.

30. Analyze legal and ethical issues related to computers.

Includes legal and ethical issues of professional conduct in the computer industry; legal issues related to copyright in the computer field; and ethical issues concerning the use of computers in contemporary society.

31. Analyze computer applications in business and industry.

Includes the uses of computers in data processing, the uses of computers in accounting and financial modeling, computer applications in industry and manufacturing, and the uses of telecommunication.

32. Identify computer applications in science, health, and the arts.

Includes the applications of computers using simulations and modeling, areas of research in artificial intelligence (e.g., heuristic programming, expert systems, robotics, cybernetics), and computer applications in the health field.

33. Analyze career opportunities involving computer science.

Includes types of careers involving computer science, the requirements of careers in computer science, and the impact of social and technological change on computer professions.

34. Understand issues in computer education.

Includes basic issues of computer literacy; and the goals and objectives of various computer curricula.

35. Analyze instructional uses of computers.

Includes the uses of computer-assisted instruction, types and characteristics of computer-assisted instructional models, the uses of computer-managed instruction, and ways to use computers in specific subject areas.

36. Analyze the uses of computers in special education.

Includes types and characteristics of students with special educational needs, types and characteristics of computerized devices used in special education, instructional activities involving computers for students with special educational needs, and the benefits of computers in special education.

37. Identify the professional responsibilities of computer science teachers.

Includes ways to evaluate, select, and develop instructional materials that involve using computers; ways to assist the school in evaluating, selecting, and acquiring computer equipment; criteria for evaluating the computer science program; and the functions and characteristics of professional organizations (e.g., Association of Computing Machinery, International Society for Technology in Education, Illinois Computing Educators) and publications involved in computer science.

38. Understand principles of measurement and evaluation as applied to instruction, assessment, and program evaluation.

Includes major terms and concepts associated with educational measurement and evaluation (e.g., reliability, validity, measurement error, norm referenced) and statistical principles needed for the interpretation of educational test results.

39. Understand how to select and develop fair, effective, and appropriate educational assessment instruments.

Includes criteria and procedures for the selection of educational tests; the creation of measurable test objectives; the uses and limitations of multiple-choice, constructed response, and performance assessments; the relationship among assessment, curriculum, and instruction; and principles of nondiscriminatory test construction.

40. Understand how to administer, score, and interpret a variety of educational assessment instruments.

Includes test administration procedures, including the preparation of testing accommodations for special populations; test scoring procedures; the use of various guides for scoring assessment instruments (e.g., scoring rubrics for essay questions, scoring scales for rating performance assessments); and the interpretation of commonly reported test scores (e.g., percentile ranks, percentile band scores, grade equivalents).

41. Understand how to use assessment data and information to promote student achievement as it relates to educational planning and school improvement.

Includes the use of assessment data to identify individual student strengths and weaknesses, the role of assessment results in program evaluation and instructional planning, and techniques for communicating student assessment data to parents/guardians, staff, and the community.

COMPUTER SCIENCE PRACTICE TEST QUESTIONS

1. In a microcomputer, which of the following stores all the information associated with software that is currently being executed?

   A. RAM

   B. hard disk

   C. ROM

   D. data register

2. In a computer, information to be transmitted to a peripheral device is gathered in the computer's:

   A. central processor.

   B. accumulator.

   C. output buffer.

   D. interface.

3. A local business has donated six personal computers to the Jefferson Middle School along with an Ethernet cable and a software package that allows the six computers to exchange e-mail and share files without the need for a central file server. To connect the computers, the school will also need to obtain one:

   A. high-speed modem for each computer.

   B. network interface card for each computer.

   C. network bridge to connect the six computers.

   D. network router to connect the six computers.

4. Which of the following would be the most appropriate type of software to select for interpreting business data?

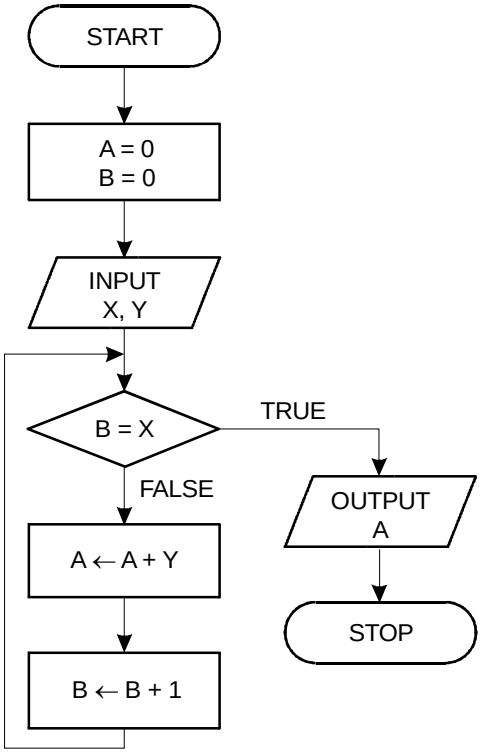
   A. CAD/CAM

   B. spreadsheet

   C. CMI

   D. telecommunications

5. **Use the diagram below to answer the question that follows.**



The diamond symbol surrounding the statement *B = X* indicates that this is:

A. a conditional statement.

B. an input statement.

C. a data processing statement.

D. a declaration statement.

6. A heuristic would most likely be used in a section of program code designed to:

A. ensure a noncatastrophic termination for all states and conditions.

B. perform complex calculations with a high level of precision.

C. isolate a large data structure from all external contacts.

D. solve a complex problem using incomplete information and inference.

7. The most basic requirement of any computer program test plan is that it should allow:

A. every loop to run to its limit.

B. all possible input values to be supplied.

C. every line of code to be executed.

D. all possible output values to be generated.

8. In programming, mnemonic identifiers are used to make a program's code:

   A. more functional.

   B. self-documenting.

   C. more reliable.

   D. user friendly.

9. In a high-level language, variable declarations are used to assign each variable a:

   A. range of potential data types.

   B. specific line of program use.

   C. maximum number of value changes.

   D. unique and constant identifier.

10. Which of the following is a unary operator?

    A. NOT

    B. <

    C. DIV

    D. /

11. **Use the pseudocode below to answer the question that follows.**

```
FUNCTION FACTORIAL (integer X): integer
    if X equals zero
        then FACTORIAL equals one
        else FACTORIAL equals X times FACTORIAL
END FACTORIAL
```

Which of the following must be added to the pseudocode above to ensure that the recursive control structure employed works properly?

A. a base case

B. a loop-counting expression

C. a general case

D. a limiting expression

12. **Use the pseudocode below to answer the question that follows.**

```
1  PROGRAM BANK_TELLER
2  BEGIN BANK_TELLER
3      GET ACCOUNT NUMBER
       AND TRANSACTION DATA
4      PERFORM CALCULATIONS
5      UPDATE ACCOUNT
       INFORMATION DATABASE
6      PRINT TRANSACTION
       RECEIPT
7  END BANK_TELLER
```

Which line in this program functions as an input procedure?

A. line 3

B. line 4

C. line 5

D. line 6

13. PASCAL is most commonly used in education to teach students how to write which of the following types of programs?

A. compiler

B. source

C. assembler

D. object

14. One of the programming features of BASIC is that program variables:

A. cannot have mnemonic identifiers.

B. are defined as they are used.

C. cannot be used as parameters.

D. are allocated dynamically.

15. During program execution, which of the following data structures is most commonly used to maintain the order and memory location of command statements?

A. linked list

B. queue

C. tree

D. stack

16. The increasing use of computers in U.S. society has caused concern about citizens' rights primarily because of the fear that computerized information may be:

A. accessed without proper authorization.

B. entered incorrectly.

C. retained after it has become obsolete.

D. purged in a system failure.

17. In accounting and finance, micro-computers are most often used to:

A. identify the best data analysis technique in a given situation.

B. gain access to the data of competitive businesses.

C. ensure the error-free entry of data into business records.

D. manipulate given data for a variety of purposes.

18. After completing a course in basic computer literacy, an individual would most likely be able to:

A. write simple computer programs.

B. identify the language used in a program code.

C. define common computer terms.

D. use major types of available software.

19. Computer-assisted instruction is most likely to benefit a classroom teacher by:

    A. decreasing the administrative work necessary for managing a classroom.

    B. increasing the time available for providing students with individualized instruction.

    C. reducing the amount of testing required to assess students' skills.

    D. ensuring that all students are working and progressing at a similar rate.

20. A sixth grade student attains a grade equivalent score of 8.6 on a sixth grade standardized reading achievement test administered in February. A correct interpretation of this score is that the sixth grader:

    A. achieved a score that eighth graders would be expected to achieve on a grade-appropriate standardized reading test administered in April.

    B. can do eighth grade work as well as the average eighth grade student in the sixth month.

    C. performed as well as students at grade level 8.6 would perform if they took the same test that the sixth grader took.

    D. reads as well as 80 percent of eighth graders.

ANSWER KEY

This section contains the answers to the practice test questions in the previous section.

After you have worked through the practice test questions, check the answers given in this section to see which questions you answered correctly.

Question Number	Correct Response	Test Objective
1.	A	Identify the general architectural characteristics of computers.
2.	C	Identify types and characteristics of peripheral devices.
3.	B	Understand the general principles of network connectivity.
4.	B	Identify types of programming software and software applications.
5.	A	Understand the function of flowcharts and pseudocode in program preparation.
6.	D	Analyze design strategies in program development.
7.	C	Understand the principles of testing computer programs.
8.	B	Identify principles of program documentation.
9.	D	Analyze declarations and data types common to high-level languages using a pseudolanguage.
10.	A	Understand statements and operators common to high-level languages using a pseudolanguage.
11.	D	Understand control structures common to high-level languages using a pseudolanguage.
12.	A	Identify input and output procedures common to high-level languages using a pseudolanguage.
13.	B	Identify and analyze features and uses of common programming languages.
14.	B	Identify and analyze features and uses of common programming languages.
15.	D	Identify the functions and uses of data structures.
16.	A	Analyze legal and ethical issues related to computers.
17.	D	Analyze computer applications in business and industry.
18.	C	Understand issues in computer education.
19.	B	Analyze instructional uses of computers.
20.	C	Understand how to administer, score, and interpret a variety of educational assessment instruments.

EXPLANATION OF THE TEST SCORE REPORT

OVERVIEW

The score report indicates whether or not you passed the test and how you performed on each test subarea. The passing scores for the Illinois Licensure Testing System were established by the Illinois State Board of Education based on recommendations from panels of Illinois educators. The passing score for each content-area test is designed to reflect the level of content knowledge and skills required to perform the job of an educator receiving an initial license in Illinois.

Passing Score

To pass the Computer Science test you must obtain a scaled total test score of 240 or above.

Total Test Score

The total test score is based on your performance on the entire test, specifically the number of multiple-choice questions you answered correctly.

Subarea Scores

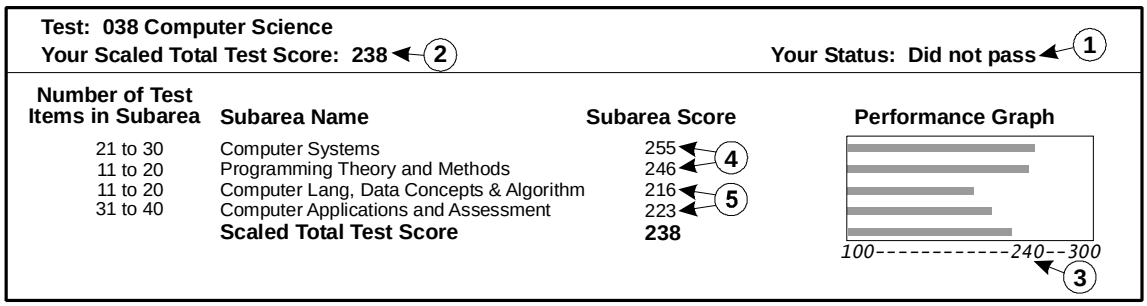
- Subarea scores are presented on the same scale as the total test score.
- Subarea scores contain different numbers of questions and are weighted differently in the computation of the total test score; therefore, the average of the subarea scaled scores generally will not equal the scaled total test score.
- Subarea scores will help you assess your areas of relative strength and weakness.

Reporting of Scores

Your results will be forwarded to the Illinois State Board of Education and to the Illinois institution(s) you indicate during the registration process. You should keep the score report you receive for your own records.

READING YOUR REPORT: A SAMPLE

A sample of a Computer Science test score report is provided below.



Test: 038 Computer Science
Your Scaled Total Test Score: 238 ← (2)
Your Status: Did not pass ← (1)

Number of Test Items in Subarea	Subarea Name	Subarea Score
21 to 30	Computer Systems	255 ← (4)
11 to 20	Programming Theory and Methods	246 ← (4)
11 to 20	Computer Lang, Data Concepts & Algorithm	216 ← (5)
31 to 40	Computer Applications and Assessment	223 ← (5)
	Scaled Total Test Score	**238**

Performance Graph

100----------240--300 ← (3)

According to the above sample, the examinee did not pass the test ①. The examinee scored 238 ② on the total test and therefore did not meet the minimum passing score of 240 ③. The examinee performed well on two subareas: Computer Systems (score of 255) ④ and Programming Theory and Methods (score of 246). However, the examinee did not perform well on two subareas: Computer Languages, Data Concepts, and Algorithms (score of 216) ⑤ and Computer Applications and Assessment (score of 223). In studying to retake the test, the examinee may wish to concentrate on studying for these two subareas.