



Michigan

TEST FOR TEACHER CERTIFICATION
STUDY GUIDE

50 Computer Science

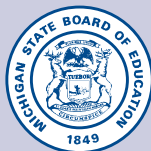


TABLE OF CONTENTS

PART 1: General Information About the MTTC Program and Test Preparation

OVERVIEW OF THE TESTING PROGRAM	1-1
Contact Information	
Test Development Process	
Characteristics of the Tests	
Test Administration	
Score Reporting	
HOW TO PREPARE FOR THE TESTS	1-4
Plan Your Course of Study	
THE DAY OF THE TEST: HELPFUL HINTS.....	1-5
Preparing for the Test Administration	
Test-Taking Tips	
Test Directions	

PART 2: Test Objectives and Sample Test Questions

INTRODUCTION	2-1
TEST OBJECTIVES	2-3
SAMPLE MULTIPLE-CHOICE TEST QUESTIONS.....	2-7
ANSWER KEY FOR THE SAMPLE MULTIPLE-CHOICE TEST QUESTIONS.....	2-17

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

Copyright © 2007 Pearson Education, Inc. or its affiliate(s). All rights reserved.
National Evaluation Systems, P.O. Box 226, Amherst, MA 01004

PART 1: General Information About the MTTC Program and Test Preparation

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 MTTC Program and Test Preparation](#)

PART 2: Test Objectives and Sample Test Questions

INTRODUCTION

This section includes a list of the test objectives, immediately followed by sample test questions and an answer key for the field covered by this study guide.

Test Objectives

As noted, the test objectives are broad, conceptual statements that reflect the knowledge, skills, and understanding an entry-level teacher needs in order to teach effectively in a Michigan classroom. Each field's list of test objectives represents the **only** source of information about what a specific test will cover and, therefore, should be studied carefully.

The test objectives are organized into groups known as "subareas." These subareas define the major content areas of the test. You will find a list of subareas at the beginning of the test objective list. The percentages shown in the list of subareas indicate the approximate weighting of the subareas on the test.

Sample Multiple-Choice Test Questions

The sample multiple-choice test questions included in this section are designed to give the test-taker an introduction to the nature of the test questions included on the MTTC test for each field. The sample test questions 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 that follows the sample 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 sample test questions, you may wish to go back and review the entire list of test objectives and descriptive statements once again.

TEST OBJECTIVES

Subarea	Approximate Percentage of Questions on Test
Educational Computing and Technology Literacy	22%
Computer Systems, Data, and Algorithms	21%
Program Design and Verification	19%
Programming Language Concepts	19%
Professional Preparation	19%

EDUCATIONAL COMPUTING AND TECHNOLOGY LITERACY

Understand basic computer technology operations and concepts.

Includes appropriate use of terminology related to computer technology; characteristics and functions of basic components of computer systems, including hardware, software, and peripherals; basic procedures related to the use of computers (e.g., saving, moving, and backing up files; inputting and retrieving data); features of computer networks; the historical development of and important trends affecting the evolution of computers and related technology; and basic knowledge of the uses of computer technology in business, industry, and society.

Understand the selection, installation, management, and maintenance of computer/technology systems and software.

Includes the evaluation and selection of computer/technology systems and software for a given purpose; the configuration of computer/technology systems and related peripherals (e.g., scanners, digital cameras, video cameras); procedures for the organization, management, and security of hardware and software; and strategies for troubleshooting and maintaining various hardware/software configurations.

Understand the use of educational and productivity software.

Includes the use of utilities and word processing, desktop publishing, and graphics programs for professional applications; the use of spreadsheets for analyzing, organizing, and graphically displaying data; procedures for designing and manipulating databases and creating customized reports; the use of teacher utility and classroom management tools to design solutions for specific purposes; selection and evaluation of educational software; and features of applications that integrate word processing, database, spreadsheet, communication, and other tools.

Understand the use of telecommunications and information access resources.

Includes types, characteristics, and uses of telecommunications tools and resources, including distance learning; the use of electronic mail, Web browser applications, and Web-page creation tools for communication and for research to support instruction; and the use of automated online search tools and intelligent agents to identify and index desired information resources.

Understand the use of computers and other technologies in research, education, problem solving, and product development.

Includes basic principles of instructional design associated with the development of multimedia and hypermedia learning materials; the development of multimedia and hypermedia products; the selection of appropriate tools for communicating concepts, conducting research, and solving problems for an intended audience and purpose; the use of technology to support oral presentations; procedures for designing and publishing simple online documents; methods of conducting research and evaluating online sources of information; and strategies and tools for working on collaborative projects, team activities, and online workgroups.

COMPUTER SYSTEMS, DATA, AND ALGORITHMS**Understand basic characteristics of computer architecture.**

Includes characteristics and functions of computer components (e.g., CPU, memory, mass storage devices); machine-level data representation; data storage; the transfer of data from one location to another; steps in the machine cycle and their synchronization; and factors limiting the accuracy of numerical computations.

Understand characteristics and functions of operating systems.

Includes the importance of an operating system in coordinating a computer's activities; comparison of single-user and multiuser systems on various platforms; the principles and processes of time sharing; characteristics of and methods for managing computer components and processes (e.g., devices, memory, virtual memory, files, multitasking) through operating systems; and characteristics of the client-server model and the relationship of operating systems to this model.

Understand types and characteristics of computer networks.

Includes characteristics of local area networks (LANs) and wide area networks (WANs); various network configurations and their characteristics; the basic structure and features of the Internet; basic principles of data transfer on the Internet and the role of network protocols; and security issues related to networks and the Internet.

Understand types and applications of data structures.

Includes characteristics and uses of arrays, stacks, queues, linked lists, and binary trees; the function of pointers (e.g., in relation to dynamic data structures); characteristics of abstract data types; and principles of encapsulation and its role in maintaining data integrity.

Understand characteristics and uses of algorithms.

Includes the role of algorithms in computing; general characteristics of algorithms (e.g., correctness, efficiency, finiteness); standard algorithms (e.g., searching, sorting); problem-solving approaches; methods used to design and represent algorithms (e.g., pseudocode); characteristics and uses of iterative and recursive structures; methods of testing algorithms; and the analysis of algorithms (e.g., time-and-space trade-offs, big-O notation).

PROGRAM DESIGN AND VERIFICATION

Apply principles and procedures for designing and preparing a program.

Includes the process of analyzing a problem or application to be addressed by a program; the role of design specifications; the uses of flowcharts, schematic drawings, and pseudocode in program preparation; differences in top-down and bottom-up design methodologies; the function of modules in program design; modularization strategies; and principles of inheritance in object-oriented programming.

Apply program development and implementation procedures.

Includes steps in the programming process; characteristics of robust programs and considerations in developing robust programs with effective user interfaces; the use of libraries in programming; the purposes of programming style conventions (e.g., indenting, spacing, comments) and their appropriate application; object-based development strategies; strategies for modifying existing programs; software tools for developing programs; and steps in the process of program implementation (i.e., translation, linking, loading).

Apply program verification principles and procedures.

Includes the purpose of program testing; the design and evaluation of a test plan; common programming errors; procedures for locating errors; and methods for debugging programs.

Apply documentation and communication principles to software development.

Includes characteristics and purposes of user and system documentation of programs; the creation of clear and appropriate program documentation; the role of written and oral communication in team-based software development projects; and the application of communication skills in computer science-related contexts.

PROGRAMMING LANGUAGE CONCEPTS

Understand types and characteristics of programming languages.

Includes differences in levels of programming language (e.g., machine language); an overview of the historical development of programming languages; functions of compilers and interpreters; characteristics of various programming paradigms (e.g., imperative, functional, object-oriented); features of various types of high-level languages; and strengths and limitations of different types of languages for a given application.

Understand the characteristics and application of data types and declarations in high-level languages.

Includes characteristics and uses of variables and constants; properties and uses of data types (e.g., integer, character, Boolean); the characteristics and use of declarations to specify constants, variables, data types and structures, functions, and parameters; and the characteristics and uses of inheritance/classes.

Understand types and characteristics of statements, operators, and control structures in high-level languages.

Includes purposes of statements, operators, and control structures; types and characteristics of operators; principles of operator precedence; the role and use of assignment statements; the purpose and use of comments; and characteristics of sequential, conditional, and repetitive control structures.

Understand the characteristics and use of program modularization in high-level languages.

Includes characteristics of program modules (e.g., subprograms, objects); the difference between local and global variables; the role of parameters; parameter-passing techniques; methods of executing a module from within a program; characteristics of event-driven software; and implementation of input and output procedures.

PROFESSIONAL PREPARATION**Understand appropriate materials, methods, resources, and curricula for teaching computer science.**

Includes the identification and modeling of problem-solving strategies for computer science instruction; the uses of computers and related technologies as teaching tools for computer science instruction; the selection and use of appropriate materials and models for teaching computer science; the identification of resources to enrich the teaching of computer science; the design, development, and evaluation of laboratory activities and demonstrations for the computer science classroom; and common features of K–12 computer science curricula and their relationship to the college computer science curriculum.

Apply practices that reflect the roles and responsibilities of computer science teachers.

Includes guidance roles (e.g., career guidance, college preparation); appropriate enrichment activities for students (e.g., extracurricular computer clubs, organized competitions); and awareness of professional computer science and computer-education associations (e.g., Michigan Association for Computer-Related Technology Users in Learning [MACUL], Association for Computing Machinery [ACM], International Society for Technology in Education [ISTE]).

Apply classroom and instructional management methodologies for teaching computer science.

Includes the selection and use of appropriate materials and methods for teaching given computer science content; computer-science laboratory management skills and techniques; the development and implementation of instructional strategies to address the needs of all students (e.g., students with various learning styles, students with special needs); the selection and application of appropriate methods of assessment and evaluation; and appropriate techniques for providing students with feedback.

Understand societal issues in computer science.

Includes ethical and legal issues related to the use of computer/technology resources (e.g., acceptable use, privacy, copyright, security); the role of computers and related technology resources in promoting lifelong learning; the effects of computers and technology on individuals and society; equity issues related to technology (e.g., access, training opportunities); knowledge of gender, cultural, and ethnic issues in computer science; and resources for adaptive assistive devices for students with special needs.

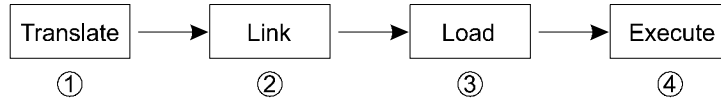
SAMPLE MULTIPLE-CHOICE TEST QUESTIONS

1. Which of the following best describes the term multitasking?
 - A. A user can have more than one application open and in use at the same time.
 - B. Processing time is managed by the operating system so that system resources are optimized.
 - C. Files and other resources are shared by users working on networked computers.
 - D. A higher-priority program can interrupt lower-priority programs to use a system device.
2. An important criterion in the evaluation of an educational software package for use in the classroom is that the program:
 - A. takes full advantage of the latest multimedia technologies.
 - B. is designed so it can be used by either individuals or groups of students.
 - C. loads quickly when it is started up.
 - D. matches clearly identified educational needs.
3. An advantage of interactive electronic learning is that:
 - A. it is less expensive than traditional classroom learning.
 - B. students can learn at their own pace and follow their own interests.
 - C. less effort is required by students to achieve success.
 - D. providing feedback to students is less labor-intensive for the teacher.
4. Which of the following steps should be taken first in the development of a Web site?
 - A. creating the links for navigating the site
 - B. drafting the text for the site
 - C. mapping out the structure and contents of the site
 - D. inserting image placeholders

5. A 2,000-byte file is edited so that its file size is increased to 6,000 bytes. The original location of the file on the hard drive does not contain enough unoccupied space around it to store the entire 6,000 bytes. Which of the following will occur when the file is saved?
- A. The surrounding files will be moved to make room for the larger file.
 - B. The file will be compressed so that it will fit into the space occupied by the original file.
 - C. The entire file will be stored in a new location on the hard drive.
 - D. The file will be stored in segments with pointers linking the nonadjacent sections.
6. The process of time-sharing on a multiuser system is achieved by:
- A. specifying the amount of time a computer has to access a peripheral device.
 - B. shuffling tasks by performing a part of each task for a specified time interval.
 - C. setting an upper limit on the amount of time a user can spend on a particular task.
 - D. allocating processing time based on the priority level of the user.
7. Which of the following accurately describes how an e-mail message is sent through the Internet to a destination in a different geographic location?
- A. It is sent directly to the destination computer with no stops in between.
 - B. It is bundled with other messages that will use the same delivery route.
 - C. It is split into packets that are sent to the destination and reassembled.
 - D. It is sent to a central distribution point that coordinates message delivery.
8. For which of the following purposes is a queue the most likely data structure to be used?
- A. storing the high and low temperatures for each day of each month in the current year
 - B. maintaining a list of jobs that are to be processed in the order in which they are received
 - C. storing inventory information where the inventory is organized by part number in ascending order
 - D. maintaining a list of book titles and authors so that a list of all books by a particular author can be obtained

9. A programmer uses a bubble sort algorithm to sort a list of 100,000 names into alphabetical order. Rather than physically swapping the data, the programmer employs a technique that uses an array of pointers that point to the names in the list and then reassigns the pointers' values as the names are sorted. The programmer has determined that this technique will save time during execution. A significant trade-off in using this technique is the increase in the:
- A. number of iterations needed.
 - B. amount of memory required during program execution.
 - C. amount of program documentation that must be written.
 - D. time required to test the program adequately.
10. A programmer is designing a program to track a department store's inventory. Which of the following actions during the program's design phase would best demonstrate that the programmer understands the function of modules?
- A. using pseudocode to write the sections of the main program that will handle input and output
 - B. deciding the order in which the algorithms will be coded
 - C. identifying and finding solutions for the individual processes necessary to track the inventory
 - D. determining the efficiency of the algorithms used in the program

11. Use the diagram below to answer the question that follows.



A program is compiled and run according to the steps in the diagram above. After the program has executed successfully, it is run again with new input but with no changes to the source code. At which step will the process begin the second time the program is executed?

- A. Step 1
- B. Step 2
- C. Step 3
- D. Step 4

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

```
BEGIN PROGRAM AVERAGE
  SET SUM = 0
  SET J = 0
  DO WHILE J < 3
    DISPLAY ("Enter a number between 1 and 100")
    ACCEPT NUM
    IF (NUM >= 1 AND <= 100)
      THEN SUM = SUM + NUM
    END IF
  END DO WHILE
  AVG = SUM / J
END PROGRAM AVERAGE
```

The pseudocode above is written to calculate the average of three integers between 1 and 100. With inputs of -1, 0, 4, 8, and 12, the program will:

- A. display the correct average.
- B. be stuck in an infinite loop.
- C. generate a run-time error.
- D. display a number that is incorrect.

13. An important purpose of system documentation for a software program is to:
- A. describe the internal structure of the software so that it can be maintained through its life cycle.
 - B. report the results of testing and debugging procedures performed on the software.
 - C. list the source code for the major modules of the program.
 - D. provide instructions to the user on how to access and revise the program's code.
14. Which of the following functions is performed by a compiler?
- A. formatting program output
 - B. evaluating the data passed to program modules
 - C. connecting multiple object programs
 - D. translating a source program into object code

15. Use the information below to answer the question that follows.

An object-oriented computer program is written to manage a simple automatic teller machine (ATM). Each customer can have a checking and a savings account. The functions listed in the table below can be performed on each type of account.

Accounts	Functions
Checking	Deposit Funds
	Withdraw Funds
Savings	Transfer Funds
	Print Account Balances

A class called "money_manage" is defined that contains the functions listed above. Which of the following will be an object of type "money_manage"?

- A. the function "Withdraw Funds"
- B. a balance printout
- C. the account "Savings"
- D. a recorded deposit

16. In a computer program, conditional control structures are used to:
- A. alter the execution sequence of the program.
 - B. verify the values that are returned from procedures.
 - C. ensure that data type declarations are made in the correct order.
 - D. store the values of arguments being passed to procedures.
17. In a program module such as a procedure or a function, the role of parameters is to:
- A. serve as the module's input and output.
 - B. indicate the number of variables that will be declared in the module.
 - C. isolate calculations that could potentially cause errors.
 - D. reassign data types to variables declared in the main program.
18. A computer science teacher would like to include instruction on a relatively new computer science topic but has had difficulty finding instructional materials. Which of the following sources would likely be most useful in terms of getting suggestions for materials to use?
- A. a publisher of computer science textbooks
 - B. an online newsgroup for computer science teachers
 - C. a national conference for computer professionals working on the topic
 - D. a professional journal for instructional technology teachers

19. A computer science teacher is planning a lab activity in which students will explore and compare features of three different operating systems. There are only a limited number of stations in the computer lab for two of the operating systems, however. Which of the following strategies is likely to be the most effective way for the teacher to manage this activity?
- A. assigning students to groups of three and having each student explore a different operating system and then share their findings with the others
 - B. scheduling the students to perform the lab activity at different times during the term so that only a few are working on it at any one time
 - C. having students learn about the operating systems through whole-class demonstrations instead of through individual hands-on work
 - D. having students rotate among the stations so that they all have an equal opportunity to work with those operating systems
20. The development of the World Wide Web has had which of the following important consequences for individuals?
- A. It has made individuals much more critical consumers of information.
 - B. It has given individuals access to a much greater amount and diversity of information.
 - C. It has allowed individuals to have greater control over who has access to their personal information.
 - D. It has improved individuals' communication skills.

ANSWER KEY FOR THE SAMPLE MULTIPLE-CHOICE TEST QUESTIONS

Item Number	Correct Response	Objective
1.	A	Understand basic computer technology operations and concepts.
2.	D	Understand the use of educational and productivity software.
3.	B	Understand the use of telecommunications and information access resources.
4.	C	Understand the use of computers and other technologies in research, education, problem solving, and product development.
5.	D	Understand basic characteristics of computer architecture.
6.	B	Understand characteristics and functions of operating systems.
7.	C	Understand types and characteristics of computer networks.
8.	B	Understand types and applications of data structures.
9.	B	Understand characteristics and uses of algorithms.
10.	C	Apply principles and procedures for designing and preparing a program.
11.	C	Apply program development and implementation procedures.
12.	B	Apply program verification principles and procedures.
13.	A	Apply documentation and communication principles to software development.
14.	D	Understand types and characteristics of programming languages.
15.	C	Understand the characteristics and application of data types and declarations in high-level languages.
16.	A	Understand types and characteristics of statements, operators, and control structures in high-level languages.
17.	A	Understand the characteristics and use of program modularization in high-level languages.
18.	B	Understand appropriate materials, methods, resources, and curricula for teaching computer science.
19.	D	Apply classroom and instructional management methodologies for teaching computer science.
20.	B	Understand societal issues in computer science.