

Chapter 01: An Overview of Computers and Programming

True / False

1. Software can be classified into two broad types: application software and programming software.

- a. True
- b. False

ANSWER: False

2. Every programming language has rules governing its word usage and punctuation.

- a. True
- b. False

ANSWER: True

3. Besides the popular, comprehensive programming languages such as Java and C++, many programmers use scripting languages such as Python, Lua, Perl, and PHP.

- a. True
- b. False

ANSWER: True

4. Professional computer programmers write programs to satisfy their own needs.

- a. True
- b. False

ANSWER: False

5. The heart of the programming process lies in planning the program's logic.

- a. True
- b. False

ANSWER: True

6. Once a program is completely coded, it is ready for a company or organization to use.

- a. True
- b. False

ANSWER: False

7. A flowchart is an English-like representation of the logical steps necessary to solve a problem.

- a. True
- b. False

ANSWER: False

Chapter 01: An Overview of Computers and Programming

8. An infinite loop is a flow of program logic that repeats and never ends.

- a. True
- b. False

ANSWER: True

9. Alan Turing is often regarded as the first programmer.

- a. True
- b. False

ANSWER: False

10. Programmers can use either procedural programming or object-oriented programming to develop programs.

- a. True
- b. False

ANSWER: True

Multiple Choice

11. ____ data items may involve organizing or sorting them, checking them for accuracy, or performing calculations with them.

- a. Processing b. Inputting
- c. Outputting d. Converting

ANSWER: a

12. ____ errors are relatively easy to locate and correct because the compiler or interpreter you use highlights every error.

- a. Logic b. Syntax
- c. Input d. Process

ANSWER: b

13. A ____ error results when you use a syntactically correct statement but use the wrong one for the current context.

- a. syntax b. logical
- c. semantic d. programming

ANSWER: b

14. A(n) ____ is a named memory location whose value can vary.

- a. input b. variable
- c. output d. logic

ANSWER: b

Chapter 01: An Overview of Computers and Programming

15. Computer programmers often refer to memory addresses using ____ notation.

- a. binary b. indirect
- c. mathematical d. hexadecimal

ANSWER: d

16. Before a programmer plans the logic of the program, he or she must ____.

- a. understand the problem b. write the program code
- c. test the program d. maintain the program

ANSWER: a

17. After a programmer plans the logic of a program, the next step is ____.

- a. understanding the problem b. testing the program
- c. translating the program d. coding the program

ANSWER: d

18. The process of walking through a program's logic on paper before you actually write the program is called ____.

- a. desk-checking b. flowcharting
- c. pseudocoding d. testing

ANSWER: a

19. Typically, a programmer develops a program's logic, writes the code, and ____ the program, receiving a list of syntax errors.

- a. runs b. compiles
- c. executes d. tests

ANSWER: b

20. The process of finding and correcting program errors is called ____.

- a. desk-checking b. syntax-checking
- c. error correcting d. debugging

ANSWER: d

21. After programs are put into production, making necessary changes is called ____.

- a. debugging b. conversion
- c. maintenance d. testing

ANSWER: c

22. Using ____ involves writing down all the steps you will use in a program.

- a. a compiler b. an interpreter
- c. a flowchart d. pseudocode

ANSWER: d

Chapter 01: An Overview of Computers and Programming

23. In a flowchart, the ____ is used to represent processing.

- a. circle b. diamond
- c. triangle d. rectangle

ANSWER: d

24. The ____ is used to represent output in a flowchart.

- a. square b. circle
- c. parallelogram d. triangle

ANSWER: c

25. The ____ is the standard terminal symbol for a flowchart.

- a. circle b. lozenge
- c. diamond d. square

ANSWER: b

26. The repetition of a series of steps is called a(n) ____.

- a. repeat flow b. loop
- c. flow d. infinite loop

ANSWER: b

27. A(n) ____ is a repeating flow of logic with no end.

- a. nonterminated condition b. infinite loop
- c. variable d. decision symbol

ANSWER: b

28. You represent a decision in a flowchart by drawing a decision symbol, which is shaped like a ____.

- a. parallelogram b. square
- c. diamond d. circle

ANSWER: c

29. Many programming languages use the term ____ to refer to the marker that is used to automatically recognize the end of data in a file.

- a. data_end b. end_data
- c. eof d. eod

ANSWER: c

Chapter 01: An Overview of Computers and Programming

30. To enter the program into a computer so you can translate and execute it, you usually use a keyboard to type program statements into a(n) ____.
- a. command line b. input screen
 - c. editor d. word processor

ANSWER: c

31. A(n) ____ is a program that you use to create simple text files.
- a. text editor b. IDE
 - c. GUI d. GDE

ANSWER: a

32. A(n) ____ is a software package that provides an editor, a compiler, and other programming tools.
- a. GUI b. CGI
 - c. GDE d. IDE

ANSWER: d

33. A(n) ____ is a location on your computer screen where you type text entries to communicate with the computer's operating system.
- a. input line b. communication line
 - c. command line d. GCI

ANSWER: c

34. A ____ allows users to interact with a program in a graphical environment.
- a. GCI b. CGI
 - c. command line d. GUI

ANSWER: d

35. The major difference between the two main programming styles in use today is the ____.
- a. use of flowcharts versus pseudocode
 - b. testing procedure used by the programmer
 - c. programmer's focus during the earliest planning stages of a project
 - d. programming language used

ANSWER: c

Completion

36. When instructions are carried out, a program runs, or _____.

ANSWER: executes

37. A(n) _____ is the sequence of steps necessary to solve any problem.

ANSWER: algorithm

Name: _____ Class: _____ Date: _____

Chapter 01: An Overview of Computers and Programming

38. When you draw a flowchart, you use a(n) _____ to represent the input symbol, which indicates an input operation.

ANSWER: parallelogram

39. A preselected value that stops the execution of a program is often called a(n) _____ value because it does not represent real data.

ANSWER: dummy

40. A(n) _____ programmer would approach a problem by dividing the process into manageable subtasks.

ANSWER: procedural

Chapter 01: An Overview of Computers and Programming

Matching

Match each term with a statement below.

- a. volatile memory
- b. software
- c. syntax error
- d. machine language
- e. sentinel
- f. pseudocode
- g. hardware
- h. flowchart
- i. flowlines
- j. documentation

41. Equipment, or the physical devices, associated with a computer

ANSWER: g

42. Instructions that tell the computer what to do

ANSWER: b

43. Contents are lost when the computer is turned off or loses power

ANSWER: a

44. Represent(s) the millions of on/off circuits within the computer

ANSWER: d

45. Incorrectly spelled words, or reversing the proper order of two words in a computer program

ANSWER: c

46. All the supporting paperwork for a program

ANSWER: j

47. Pictorial representation of the logical steps it takes to solve a problem

ANSWER: h

48. English-like representation of the logical steps it takes to solve a problem

ANSWER: f

49. Used to show the correct sequence of statements

ANSWER: i

50. Preselected value that stops the execution of a program

ANSWER: e

Chapter 01: An Overview of Computers and Programming

Subjective Short Answer

51. Describe internal and permanent storage.

ANSWER: When you type program instructions, they are stored in computer memory, which is a computer's temporary, internal storage. Internal storage is volatile—its contents are lost when the computer is turned off or loses power. Usually, you want to be able to retrieve and perhaps modify the stored instructions later, so you also store them on a permanent storage device, such as a disk. Permanent storage devices are nonvolatile—that is, their contents are persistent and are retained even when power is lost.

52. Describe the function of a compiler and an interpreter.

ANSWER: Each programming language uses a piece of software, called a compiler or an interpreter, to translate your program code into machine language. Machine language is also called binary language, and is represented as a series of 0s and 1s. The compiler or interpreter that translates your code tells you if any programming language component has been used incorrectly. Syntax errors are relatively easy to locate and correct because the compiler or interpreter you use highlights every syntax error. If you write a computer program using a language such as C++ but spell one of its words incorrectly or reverse the proper order of two words, the software lets you know that it found a mistake by displaying an error message as soon as you try to translate the program.

53. What is a variable and how is it used in a program?

ANSWER: Suppose the location `myNumber` is a variable. A variable is a named memory location whose value can vary—for example, the value of `myNumber` might be 3 when the program is used for the first time, and 45 when it is used the next time.

54. What is involved in putting a program into production?

ANSWER: Once the program is tested adequately, it is ready for the organization to use. Putting the program into production might mean simply running the program once, if it was written to satisfy a user's request for a special list. However, the process might take months if the program will be run on a regular basis, or if it is one of a large system of programs being developed. Perhaps data-entry people must be trained to prepare the input for the new program; users must be trained to understand the output; or existing data in the company must be changed to an entirely new format to accommodate this program. Conversion, the entire set of actions an organization must take to switch over to using a new program or set of programs, can sometimes take months or years to accomplish.

55. How is a decision represented in a flowchart?

ANSWER: You represent a decision in a flowchart by drawing a decision symbol, which is shaped like a diamond. The diamond usually contains a question, the answer to which is one of two mutually exclusive options—often yes or no. All good computer questions have only two mutually exclusive answers, such as yes and no, or true and false. For example, “What day of the year is your birthday?” is not a good computer question because there are 366 possible answers. But “Is your birthday June 24?” is a good computer question because, for everyone in the world, the answer is either yes or no.

Chapter 01: An Overview of Computers and Programming

56. How does a program recognize the end of an input file?

ANSWER: Programming languages can recognize the end of data in a file automatically, through a code that is stored at the end of the data. Many programming languages use the term `eof` (for “end of file”) to refer to this marker that automatically acts as a sentinel.

57. Discuss two ways that you can enter a program into a computer.

ANSWER: To enter the program into a computer so you can translate and execute it, you usually use a keyboard to type program statements into an editor. You can type a program into one of the following:

A plain text editor

A text editor that is part of an integrated development environment

A text editor is a program that you use to create simple text files. It is similar to a word processor, but without as many features. You can use a text editor such as Notepad that is included with Microsoft Windows. An advantage to using a simple text editor to type and save a program is that the completed program does not require much disk space for storage.

You can use the editor of an integrated development environment (IDE) to enter your program. An IDE is a software package that provides an editor, a compiler, and other programming tools.

58. List the features that are commonly included with an IDE editor.

ANSWER: It uses different colors to display various language components, making elements like data types easier to identify.

It highlights syntax errors visually for you.

It employs automatic statement completion; when you start to type a statement, the IDE suggests a likely completion, which you can accept with a keystroke.

It provides tools that allow you to step through a program’s execution one statement at a time so you can more easily follow the program’s logic and determine the source of any errors.

Chapter 01: An Overview of Computers and Programming

59. What are some of the features provided by modern programming languages that were not available in older programming languages?

ANSWER: People have been writing modern computer programs since the 1940s. The oldest programming languages required programmers to work with memory addresses and to memorize awkward codes associated with machine languages. Newer programming languages look much more like natural language and are easier to use, partly because they allow programmers to name variables instead of using awkward memory addresses. Also, newer programming languages allow programmers to create self-contained modules or program segments that can be pieced together in a variety of ways. The oldest computer programs were written in one piece, from start to finish, but modern programs are rarely written that way—they are created by teams of programmers, each developing reusable and connectable program procedures. Writing several small modules is easier than writing one large program, and most large tasks are easier when you break the work into units and get other workers to help with some of the units.

60. Currently, there are two major techniques used to develop programs and their procedures. Name and describe them.

ANSWER: One technique, procedural programming, focuses on the procedures that programmers create. In other words, procedural programmers focus on the actions that are carried out—for example, getting input data for an employee and writing the calculations needed to produce a paycheck from the data. Procedural programmers would approach the job of producing a paycheck by breaking down the paycheck-producing process into manageable subtasks.

The other popular programming model, object-oriented programming, focuses on objects, or “things,” and describes their features (or attributes) and their behaviors. For example, object-oriented programmers might design a payroll application by thinking about employees and paychecks, and describing their attributes (e.g., employees have names and Social Security numbers, and paychecks have names and check amounts).