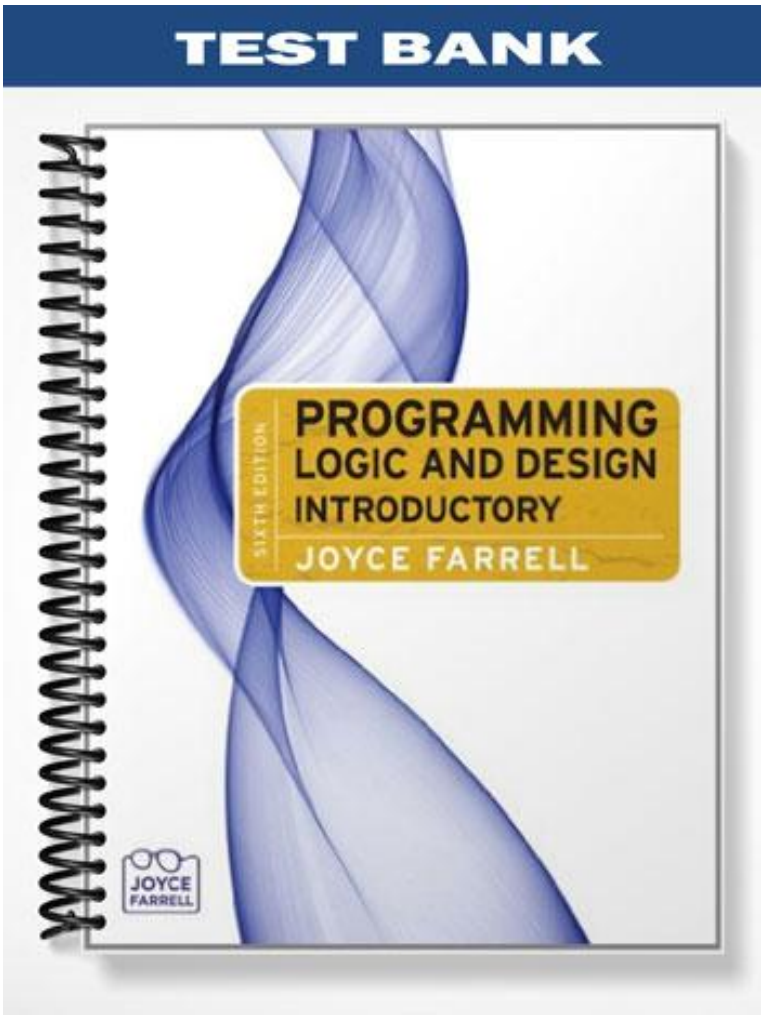
TEST BANK
PROGRAMMING
LOGIC AND DESIGN
INTRODUCTORY
SIXTH EDITION
JOYCE FARRELL
JOYCE
FARRELL

ch02

True/False
Indicate whether the statement is true or false.

____ 1. At any moment in time, a variable can hold more than one value.

____ 2. The ability of variables to change in value is what makes computers and programming worthwhile.

____ 3. In many programming languages, if you declare a variable and do not initialize it, the variable contains an unknown value until it is assigned a value.

____ 4. Programmers generally write programs as one long series of steps.

____ 5. Most modern programming languages require that program statements be placed in specific columns.

Multiple Choice
Identify the choice that best completes the statement or answers the question.

____ 6. When you write programs, you work with data in three different forms: ____.
a. values; variables, or named values; and unnamed values
b. variables; named constants; and named memory
c. variables; literals, or unnamed constants; and named constants
d. variations; transliterals, or unnamed constants; and named values

____ 7. In most programming languages, before you can use any variable, you must include a ____ for it.
a. declaration
b. definition
c. header
d. proclamation

____ 8. The process of naming program variables and assigning a type to them is called ____ variables.
a. initializing
b. declaring
c. identifying
d. proclaiming

____ 9. A variable's unknown value is commonly called ____.
a. initial
b. default
c. deterministically random
d. garbage

____ 10. You can also refer to a variable name as a ____.
a. mnemonic
b. pneumonic
c. cue
d. prompt

____ 11. When the variable starts with a lowercase letter and any subsequent word begins with an uppercase letter, this is called ____.
a. Hungarian notation
b. Pascal
c. camel casing
d. Turing notation

____ 12. When the first letter of a variable name is uppercase, as in `HourlyWage`, the format is known as ____ casing.
a. Hungarian notation
b. Pascal
c. camel casing
d. Turing notation

____ 13. A specific numeric value is often called a ____.
a. named constant
b. defined constant
c. arithmetic constant
d. numeric constant

____ 14. Fractional numeric variables that contain a decimal point are known as ____ variables.
a. partial
b. string
c. integer
d. floating-point

____ 15. A(n) ____ is similar to a variable, except it can be assigned a value only once.
a. unnamed constant
b. literal
c. named constant
d. constant

____ 16. The ____ dictate the order in which operations in the same statement are carried out.
a. rules of precedence
b. statement rules
c. operation rules
d. rules of arithmetic

____ 17. The process of breaking down a large program into modules is called ____.
a. decomposition
b. modularization
c. unification
d. orientation

____ 18. ____ is the process of paying attention to important properties while ignoring nonessential details.
a. Abstraction
b. Modularization
c. Abbreviation
d. Decomposition

____ 19. Programmers say the statements that are contained in a module have been ____.
a. embedded
b. decomposed
c. encapsulated
d. modularized

____ 20. Programmers say that variables and constants declared within a module are ____ only within that module.
a. abstracted
b. out of scope
c. in scope
d. in reference

____ 21. ____ variables and constants are known to the entire program.
a. Local
b. Transient
c. Heap
d. Global

____ 22. When a program has several modules calling other modules, programmers often use a program ____, which operates similarly to an organizational chart, to show the overall picture of how modules are related to one another.
a. hierarchy chart
b. tree chart
c. flow chart
d. data diagram

____ 23. An ____ is most often represented by a three-sided box that is connected to the step it references by a dashed line.
a. abstraction symbol
b. annotation symbol
c. abbreviation symbol
d. enumeration symbol

____ 24. Programmers refer to programs that contain meaningful names as ____.
a. undocumented
b. procedurally documented
c. formally documented
d. self-documenting

____ 25. ____ is where a variable's data type or other information is stored as part of the name.
a. Hungarian notation
b. Pascal notation
c. Turing notation
d. Camel case

____ 26. A ____ variable is not used for input or output, but instead is just a working variable that you use during a program's execution.
a. programming
b. throw away
c. temporary
d. calculating

Completion

Complete each statement.

27. Declaring a starting value is known as _____ the variable.

28. Each programming language has a few reserved _____ that are not allowed as variable names because they are part of the language's syntax.

29. Whole number numeric variables are known as _____ numeric variables.

30. _____ tasks include any steps you must perform at the beginning of a program to get ready for the rest of the program.

31. Program _____ are written explanations that are not part of the program logic but that serve as documentation for readers of the program.

Matching

Match each item with a statement below.

a. Reliability
b. Declaration
c. Echoing input
d. String variable
e. Identifier
f. Prompt
g. Variables
h. Data dictionary
i. Numeric variable

____ 32. Named memory locations whose contents can vary or differ over time
____ 33. A statement that provides a data type and an identifier for a variable
____ 34. A variable's name
____ 35. Can hold digits and have mathematical operations performed on it
____ 36. Can hold text, such as letters of the alphabet, and other special characters, such as punctuation marks
____ 37. The feature of programs that assures you a module has been tested and proven to function correctly
____ 38. A list of every variable name used in a program, along with its type, size, and description
____ 39. A message that is displayed on a monitor to ask the user for a response and perhaps explain how that response should be formatted
____ 40. The act of repeating input back to a user either in a subsequent prompt or in output

Short Answer

41. What does an item's data type describe?

42. List three reasons for modularizing a large program.

43. What items should you include when you create a module?

44. Explain the purpose of detail loop tasks.

45. What are end-of-job tasks?

46. List three design features that you can use while creating programs to make them easier to write and maintain.

47. Explain the purpose of annotation symbols.

48. Discuss why it is important to use meaningful names for identifiers.

49. Explain the purpose of temporary variables.

50. Discuss why it is important to maintain good programming habits.

ch02
Answer Section

TRUE/FALSE

1. ANS: F PTS: 1 REF: 42
2. ANS: T PTS: 1 REF: 42
3. ANS: T PTS: 1 REF: 44
4. ANS: F PTS: 1 REF: 52
5. ANS: F PTS: 1 REF: 73

MULTIPLE CHOICE

6. ANS: C PTS: 1 REF: 42
7. ANS: A PTS: 1 REF: 43
8. ANS: B PTS: 1 REF: 43
9. ANS: D PTS: 1 REF: 44
10. ANS: A PTS: 1 REF: 44
11. ANS: C PTS: 1 REF: 44
12. ANS: B PTS: 1 REF: 45
13. ANS: D PTS: 1 REF: 45
14. ANS: D PTS: 1 REF: 46
15. ANS: C PTS: 1 REF: 47
16. ANS: A PTS: 1 REF: 50
17. ANS: B PTS: 1 REF: 52
18. ANS: A PTS: 1 REF: 53
19. ANS: C PTS: 1 REF: 57
20. ANS: C PTS: 1 REF: 59
21. ANS: D PTS: 1 REF: 61
22. ANS: A PTS: 1 REF: 66
23. ANS: B PTS: 1 REF: 69
24. ANS: D PTS: 1 REF: 71
25. ANS: A PTS: 1 REF: 72
26. ANS: C PTS: 1 REF: 73

COMPLETION

27. ANS: initializing

    PTS: 1 REF: 43
28. ANS: keywords

    PTS: 1 REF: 44
29. ANS: integer

PTS: 1 REF: 46

30. ANS: Housekeeping

PTS: 1 REF: 61

31. ANS: comments

PTS: 1 REF: 69

MATCHING

32. ANS: G PTS: 1 REF: 42
33. ANS: B PTS: 1 REF: 43
34. ANS: E PTS: 1 REF: 43
35. ANS: I PTS: 1 REF: 46
36. ANS: D PTS: 1 REF: 46
37. ANS: A PTS: 1 REF: 55
38. ANS: H PTS: 1 REF: 72
39. ANS: F PTS: 1 REF: 74
40. ANS: C PTS: 1 REF: 75

SHORT ANSWER

41. ANS:
A data item's data type is a classification that describes the following:
1) What values can be held by the item
2) How the item is stored in computer memory
3) What operations can be performed on the data item

PTS: 1 REF: 43 TOP: Critical Thinking

42. ANS:
1) Modularization provides abstraction.
2) Modularization allows multiple programmers to work on a problem.
3) Modularization allows you to reuse your work more easily.

PTS: 1 REF: 52 TOP: Critical Thinking

43. ANS:
When you create a module, you include the following:
1) A header—A module's header includes the module identifier and possibly other necessary identifying information.
2) A body—A module's body contains all the statements in the module.
3) A return statement—A module's return statement marks the end of the module and identifies the point at which control returns to the program or module that called the module.

PTS: 1 REF: 55 TOP: Critical Thinking

44. ANS:
Detail loop tasks do the core work of the program. When a program processes many records, detail loop tasks execute repeatedly for each set of input data until there are no more. For example, in a payroll program, the same set of calculations is executed repeatedly until a check has been produced for each employee.

PTS: 1 REF: 62 TOP: Critical Thinking

45. ANS:
End-of-job tasks are the steps you take at the end of the program to finish the application. You can call these finish-up or clean-up tasks. They might include displaying totals or other final messages and closing any open files.

PTS: 1 REF: 62 TOP: Critical Thinking

46. ANS:
Student should list three of the following:
1) You should use program comments where appropriate.
2) Your identifiers should be well-chosen.
3) You should strive to design clear statements within your programs and modules.
4) You should write clear prompts and echo input.
5) You should continue to maintain good programming habits as you develop your programming skills.

PTS: 1 REF: 69 TOP: Critical Thinking

47. ANS:
In a flowchart, you can use an annotation symbol to hold information that expands on what is stored within another flowchart symbol. An annotation symbol is most often represented by a three-sided box that is connected to the step it references by a dashed line. Annotation symbols are used to hold comments, or sometimes statements that are too long to fit neatly into a flowchart symbol.

PTS: 1 REF: 69 TOP: Critical Thinking

48. ANS:
Creating a data item named `someData` or a module named `firstModule()` makes a program cryptic. Not only will others find it hard to read your programs, but you will forget the purpose of these identifiers even within your own programs. All programmers occasionally use short, non-descriptive names such as x or temp in a quick program; however, in most cases, data and module names should be meaningful. Programmers refer to programs that contain meaningful names as self-documenting. This means that even without further documentation, the program code explains itself to readers.

PTS: 1 REF: 71 TOP: Critical Thinking

49. ANS:
When you need several mathematical operations to determine a result, consider using a series of temporary variables to hold intermediate results. A temporary variable (or a work variable) is not used for input or output, but instead is just a working variable that you use during a program's execution.

PTS: 1 REF: 73 TOP: Critical Thinking

50. ANS:
When you learn a programming language and begin to write lines of program code, it is easy to forget the principles you have learned in this text. Having some programming knowledge and a keyboard at your fingertips can lure you into typing lines of code before you think things through. But every program you write will be better if you plan before you code. If you maintain the habit of first drawing flowcharts or writing pseudocode, as you have learned here, your future programming projects will go more smoothly. If you desk-check your program logic on paper before starting to type statements in a programming language, your programs will run correctly sooner. If you think carefully about the variable and module names you use, and design your program statements to be easy to read and use, your programs will be easier to develop and maintain.

PTS: 1 REF: 76 TOP: Critical Thinking