Chapter 2: Elements of High-Quality Programs

TRUE/FALSE

1. A variable can hold more than one value at any given moment in time.

   ANS: F PTS: 1 REF: 39

2. Because one memory location can be used repeatedly with different values, you can write program instructions once and then use them for thousands of separate calculations

   ANS: T PTS: 1 REF: 39

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.

   ANS: T PTS: 1 REF: 40

4. Variable names can be more than one word with blanks between the words.

   ANS: F PTS: 1 REF: 41

5. The assignment operator has left-to-right-to-left associativity, which means that the value of the expression to the left of the assignment operator is evaluated first and that the result is assigned to the operand on the right.

   ANS: F PTS: 1 REF: 42

6. A string variable can hold digits such as phone numbers and zip codes.

   ANS: T PTS: 1 REF: 43

7. Programmers generally write programs as one long series of steps.

   ANS: F PTS: 1 REF: 48

8. Modularization makes it harder for multiple programmers to work on a problem.

   ANS: F PTS: 1 REF: 50

9. Program comments are a type of internal documentation.

   ANS: T PTS: 1 REF: 64

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

    ANS: F PTS: 1 REF: 68

MULTIPLE CHOICE

1. 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

ANS: C PTS: 1 REF: 38

2. A specific numeric value is often called a(n) ____.
a. named constant
b. defined constant
c. arithmetic constant
d. numeric constant

ANS: D PTS: 1 REF: 38

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

ANS: D PTS: 1 REF: 38

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

ANS: A PTS: 1 REF: 39

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

ANS: B PTS: 1 REF: 39

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

ANS: D PTS: 1 REF: 40

7. Declaring a starting value for a variable is known as ____ the variable.
a. initializing
b. declaring
c. defining
d. identifying

ANS: A PTS: 1 REF: 40

8. 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 casing
c. camel casing
d. Turing notation

ANS: C PTS: 1 REF: 41

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

ANS: B PTS: 1 REF: 41

10. ____ is where a variable's data type or other information is stored as part of the name.

a. Hungarian notation
b. Pascal case
c. Turing notation
d. Camel case

ANS: A PTS: 1 REF: 41

11. The assignment operator is the ____ sign.
a. *
b. +
c. =
d. /

ANS: C PTS: 1 REF: 42

12. 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

ANS: C PTS: 1 REF: 44

13. 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

ANS: A PTS: 1 REF: 46

14. Depending on the programming language being used, modules are also known as ____ .
a. subroutines, procedures, or methods
b. subroutines, code bits, or methods
c. tasks, functions, or methods
d. procedures, functions, or hierarchy

ANS: A PTS: 1 REF: 48

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

ANS: B PTS: 1 REF: 49

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

ANS: A PTS: 1 REF: 49

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

ANS: C PTS: 1 REF: 55

18. 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

ANS: C PTS: 1 REF: 57

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

ANS: D PTS: 1 REF: 57

20. The mainline logic of almost every procedural computer program consists of these three distinct parts: ____ .
    a. housekeeping tasks, processing tasks, and end-of-job tasks
    b. clearing tasks, detail loop tasks, and end-of-job tasks
    c. housekeeping tasks, detail loop tasks, and end-of-job tasks
    d. housekeeping tasks, detail loop tasks, and math tasks

    ANS: C PTS: 1 REF: 57-58

21. 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

    ANS: A PTS: 1 REF: 61

22. As programs become larger and more complicated, the need for good planning and design ____ .
    a. decreases
    b. is inefficient
    c. is not necessary
    d. increases

    ANS: D PTS: 1 REF: 63

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

    ANS: B PTS: 1 REF: 64

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

    ANS: D PTS: 1 REF: 66

25. 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

    ANS: C PTS: 1 REF: 68

COMPLETION

1. Whole number variables are known as _____ variables.

   ANS: integer

   PTS: 1 REF: 38

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

ANS: initializing

PTS: 1 REF: 40

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

   ANS:
   keywords
   key words

   PTS: 1 REF: 41

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

   ANS:
   Housekeeping
   House-keeping
   House keeping
   housekeeping
   house-keeping
   house keeping

   PTS: 1 REF: 57

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

   ANS: comments

   PTS: 1 REF: 64

MATCHING

Match each term 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
j. Type-safety

1. Named memory locations whose contents can vary or differ over time
2. A statement that provides a data type and an identifier for a variable
3. A program component's name
4. Can hold digits and have mathematical operations performed on it
5. Can hold text, such as letters of the alphabet, and other special characters, such as punctuation marks
6. The feature of programming languages that prevents assigning values of an incorrect data type
7. The feature of programs that assures you a module has been tested and proven to function correctly
8. A list of every variable name used in a program, along with its type, size, and description
9. A message that is displayed on a monitor to ask the user for a response and perhaps explain how that response should be formatted

10. The act of repeating input back to a user either in a subsequent prompt or in output

1.	ANS: G	PTS: 1	REF: 39		
2.	ANS: B	PTS: 1	REF: 39		
3.	ANS: E	PTS: 1	REF: 39		
4.	ANS: I	PTS: 1	REF: 43		
5.	ANS: D	PTS: 1	REF: 43		
6.	ANS: J	PTS: 1	REF: 44		
7.	ANS: A	PTS: 1	REF: 50-51		
8.	ANS: H	PTS: 1	REF: 67		
9.	ANS: F	PTS: 1	REF: 69		
10.	ANS: C	PTS: 1	REF: 70		

SHORT ANSWER

1. What does a data item's data type describe?

   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: 39 TOP: Critical Thinking

2. List three reasons for modularizing a large program.

   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: 49 TOP: Critical Thinking

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

   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: 51 TOP: Critical Thinking

4. Explain the purpose of detail loop tasks.

   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: 58 TOP: Critical Thinking

5. What are end-of-job tasks?

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: 58 TOP: Critical Thinking

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

ANS:
Students 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: 63-64 TOP: Critical Thinking

7. Explain the purpose of annotation symbols.

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: 64 TOP: Critical Thinking

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

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: 66 TOP: Critical Thinking

9. Explain the purpose of temporary variables.

   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: 68 TOP: Critical Thinking

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

    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: 71 TOP: Critical Thinking