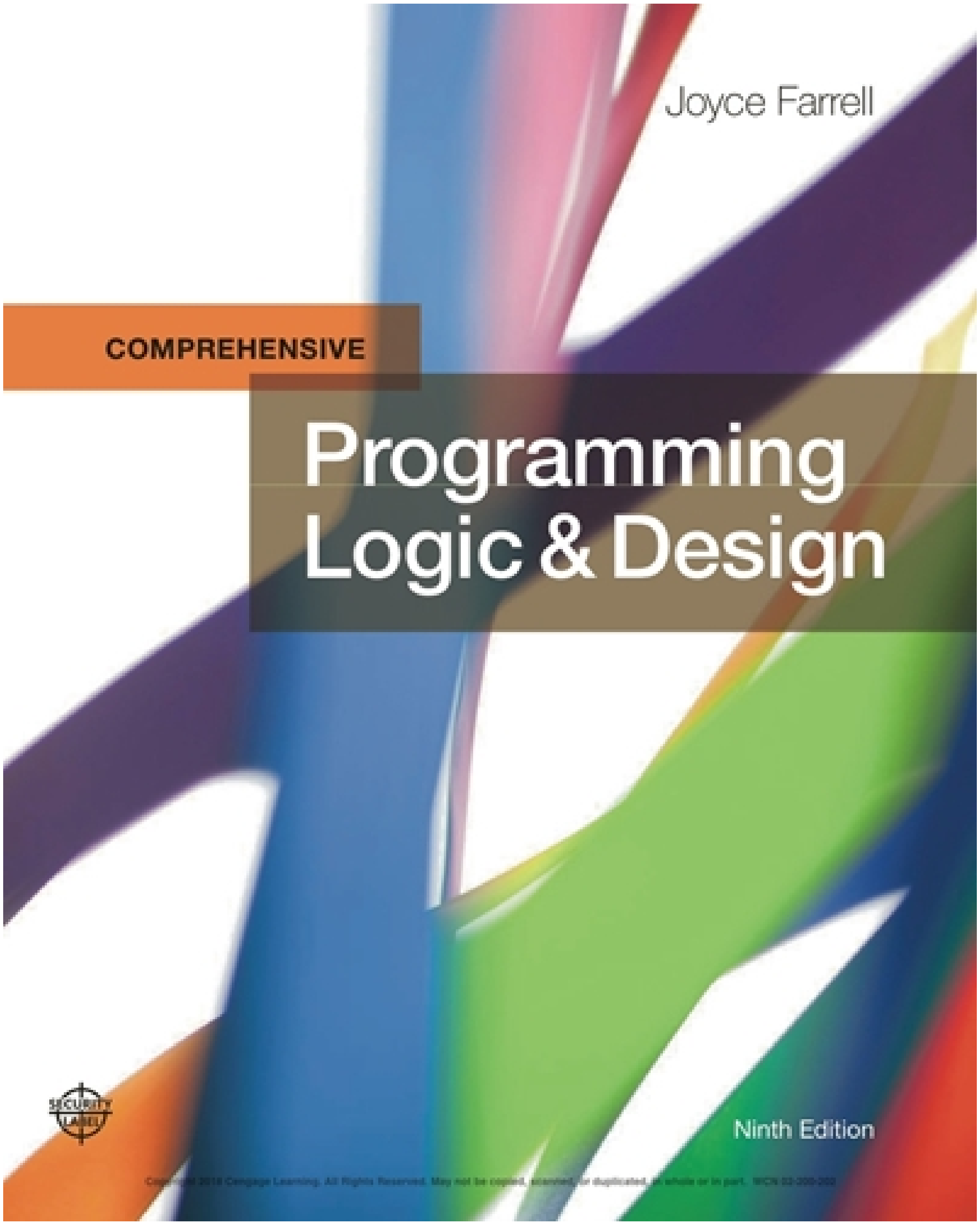
Test Bank for Programming Logic & Design Comprehensive 9th Edition by Farrell

CLICK HERE TO ACCESS COMPLETE Test Bank


Joyce Farrell
COMPREHENSIVE
Programming Logic & Design
SECURITY LABEL
Ninth Edition
Copyright 2018 Cengage Learning. All Rights Reserved. May not be copied, scanned, or duplicated, in whole or in part. WCN 02-200-202

Test Bank

Name:______________________________ Class:____________________ Date:__________

Chapter 02: Elements of High-Quality Programs

1. All programming languages support four broad data types.

   a. True

   b. False

ANSWER: False

FEEDBACK:
Correct Correct. There are two broad data types: numeric and string.
Incorrect Incorrect. There are two broad data types: numeric and string.

POINTS: 1

REFERENCES: 39
Understanding Data Types

QUESTION TYPE: True / False

HAS VARIABLES: False

LEARNING OBJECTIVES: PL&D.FARR.18.02.01 - Declare and use variables and constants

DATE CREATED: 9/12/2016 12:40 PM

DATE MODIFIED: 11/15/2019 4:11 PM

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

   a. True

   b. False

ANSWER: True

FEEDBACK:
Correct Correct. Variables are named memory locations whose contents can vary or differ over time. The ability of variables to change in value is what makes computers and programming worthwhile.
Incorrect Incorrect. Variables are named memory locations whose contents can vary or differ over time. The ability of variables to change in value is what makes computers and programming worthwhile.

POINTS: 1

REFERENCES: 40
Understanding Unnamed, Literal Constants

QUESTION TYPE: True / False

HAS VARIABLES: False

LEARNING OBJECTIVES: PL&D.FARR.18.02.01 - Declare and use variables and constants

DATE CREATED: 9/12/2016 12:40 PM

DATE MODIFIED: 11/15/2019 4:11 PM

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.

   a. True

   b. False

ANSWER: True

FEEDBACK:
Correct Correct. A variable’s unknown value commonly is called *garbage*, although some languages use a default value for some variables.
Incorrect Incorrect. A variable’s unknown value commonly is called *garbage*, although some languages use a default value for some variables.

POINTS:	1
REFERENCES:	46 Assigning Values to Variables
QUESTION TYPE:	True / False
HAS VARIABLES:	False
LEARNING OBJECTIVES:	PL&D.FARR.18.02.01 - Declare and use variables and constants
DATE CREATED:	9/12/2016 12:40 PM
DATE MODIFIED:	11/15/2019 4:11 PM

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

   a. True

   b. False

ANSWER: False

FEEDBACK:

Correct Correct. Variable names are often more than one word, but words are separated following a variety of naming conventions. Some common naming conventions include camel case, Pascal case, snake case, mixed case with underscores, and kebob case.

Incorrect Incorrect. Variable names are often more than one word, but words are separated following a variety of naming conventions. Some common naming conventions include camel case, Pascal case, snake case, mixed case with underscores, and kebob case.

POINTS:	1
REFERENCES:	43 Understanding a Declaration’s Identifier
QUESTION TYPE:	True / False
HAS VARIABLES:	False
LEARNING OBJECTIVES:	PL&D.FARR.18.02.01 - Declare and use variables and constants
DATE CREATED:	9/12/2016 12:40 PM
DATE MODIFIED:	11/15/2019 4:11 PM

5. The assignment operator has left-to-right 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.

   a. True

   b. False

ANSWER: False

FEEDBACK:

Correct Correct. The assignment operator always operates from right to left, which means that it has right-associativity or right-to-left associativity. This means that the value of the expression on the right of the assignment is evaluated first, and then the result is assigned to the operand on the left.

Incorrect Incorrect. The assignment operator always operates from right to left, which means that it has right-associativity or right-to-left associativity. This means that the value of the expression on the right of the assignment is evaluated first, and then the result is assigned to the operand on the left.

POINTS:	1
REFERENCES:	45 Assigning Values to Variables

QUESTION TYPE:	True / False
HAS VARIABLES:	False
LEARNING OBJECTIVES:	PL&D.FARR.18.02.01 - Declare and use variables and constants
DATE CREATED:	9/12/2016 12:40 PM
DATE MODIFIED:	11/15/2019 4:11 PM

6. A string variable can hold digits such as account numbers and zip codes.
   a. True
   b. False

ANSWER:	True
FEEDBACK:	*Correct* Correct. A string variable can hold text, such as letters of the alphabet, and other special characters, such as punctuation marks. Programmers frequently use strings to hold digits when they will never be used in arithmetic statements. *Incorrect* Incorrect. A string variable can hold text, such as letters of the alphabet, and other special characters, such as punctuation marks. Programmers frequently use strings to hold digits when they will never be used in arithmetic statements.
POINTS:	1
REFERENCES:	41 Understanding a Declaration’s Data Type
QUESTION TYPE:	True / False
HAS VARIABLES:	False
LEARNING OBJECTIVES:	PL&D.FARR.18.02.01 - Declare and use variables and constants
DATE CREATED:	9/12/2016 12:40 PM
DATE MODIFIED:	11/15/2019 4:11 PM

7. Programmers generally write programs as one long series of steps.
   a. True
   b. False

ANSWER:	False
FEEDBACK:	*Correct* Correct. Programmers seldom write programs as one long series of steps. Instead, they break down their programming problems into smaller units and tackle one cohesive task at a time. These smaller units are *modules*. *Incorrect* Incorrect. Programmers seldom write programs as one long series of steps. Instead, they break down their programming problems into smaller units and tackle one cohesive task at a time. These smaller units are *modules*.
POINTS:	1
REFERENCES:	51 Understanding the Advantages of Modularization
QUESTION TYPE:	True / False
HAS VARIABLES:	False
LEARNING OBJECTIVES:	PL&D.FARR.18.02.03 - Appreciate the advantages of modularization
DATE CREATED:	9/12/2016 12:40 PM
DATE MODIFIED:	11/15/2019 4:11 PM

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

a. True

b. False

ANSWER: False

FEEDBACK:

Correct Correct. When you divide any large task into modules, you gain the ability to divide the task among various people and can assign each module to an individual programmer or team.

Incorrect Incorrect. When you divide any large task into modules, you gain the ability to divide the task among various people and can assign each module to an individual programmer or team.

POINTS: 1

REFERENCES: 52
Understanding the Advantages of Modularization

QUESTION TYPE: True / False

HAS VARIABLES: False

LEARNING OBJECTIVES: PL&D.FARR.18.02.03 - Appreciate the advantages of modularization

DATE CREATED: 9/12/2016 12:40 PM

DATE MODIFIED: 11/15/2019 4:11 PM

9. Program comments are a type of internal documentation.

a. True

b. False

ANSWER: True

FEEDBACK:

Correct Correct. Program comments are written explanations that are not part of the program logic but that serve as documentation for readers of the program. In other words, they are nonexecuting statements that help readers understand programming statements.

Incorrect Incorrect. Program comments are written explanations that are not part of the program logic but that serve as documentation for readers of the program. In other words, they are nonexecuting statements that help readers understand programming statements.

POINTS: 1

REFERENCES: Using Program Comments
67

QUESTION TYPE: True / False

HAS VARIABLES: False

LEARNING OBJECTIVES: PL&D.FARR.18.02.06 - Describe the features of good program design

DATE CREATED: 9/12/2016 12:40 PM

DATE MODIFIED: 11/15/2019 4:11 PM

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

a. True

b. False

ANSWER: False

FEEDBACK:

Correct Correct. Some older programming languages require that program statements be placed in specific columns. Most modern programming languages are free-form, so you can arrange your lines of code any way you see fit.

Incorrect	Incorrect. Some older programming languages require that program statements be placed in specific columns. Most modern programming languages are free-form, so you can arrange your lines of code any way you see fit.

POINTS:	1
REFERENCES:	71 Avoiding Confusing Line Breaks
QUESTION TYPE:	True / False
HAS VARIABLES:	False
LEARNING OBJECTIVES:	PL&D.FARR.18.02.06 - Describe the features of good program design
DATE CREATED:	9/12/2016 12:40 PM
DATE MODIFIED:	11/15/2019 4:11 PM

11. 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, RAM (or unnamed constants), and named values

ANSWER:	c
FEEDBACK:	a. Incorrect. A value is neither a data type nor a form of data. b. Incorrect. Variables are named memory locations, whose contents can vary or differ over time. c. Correct. When you write programs, you work with data in two different types—numeric and string—and in three different forms: literals (unnamed constants), variables, and named constants. d. Incorrect. Variations and RAM are not forms of data.
POINTS:	1
REFERENCES:	39 Declaring and Using Variables and Constants
QUESTION TYPE:	Multiple Choice
HAS VARIABLES:	False
LEARNING OBJECTIVES:	PL&D.FARR.18.02.01 - Declare and use variables and constants
DATE CREATED:	9/12/2016 12:40 PM
DATE MODIFIED:	11/15/2019 4:11 PM

12. Fractional numeric variables that contain a decimal point are known as ____ variables.

a. partial b. string

c. integer d. floating-point

ANSWER:	d
FEEDBACK:	a. Incorrect. A partial variable is not a defined data type. b. Incorrect. A string describes data items that are nonnumeric. String data cannot be used in arithmetic operations. c. Incorrect. An integer is a whole number. d. Correct. A floating-point number is also called a real number.
POINTS:	1

REFERENCES: 39
Understanding Data Types

QUESTION TYPE: Multiple Choice

HAS VARIABLES: False

LEARNING OBJECTIVES: PL&D.FARR.18.02.01 - Declare and use variables and constants

DATE CREATED: 9/12/2016 12:40 PM

DATE MODIFIED: 11/15/2019 4:11 PM

13. A specific numeric value is often called a(n) ____.

a. named constant b. defined constant
c. arithmetic constant d. numeric constant

ANSWER: d

FEEDBACK:

a. Incorrect. A named constant is similar to a variable, except it can be assigned a value only once. You use a named constant when you want to assign a useful name for a value that will never be changed during a program's execution.

b. Incorrect. *Defined constant* is an undefined term in programming. It is likely a confusion of a named constant or the process of defining a constant.

c. Incorrect. *Arithmetic constant* is an undefined term in programming. It is a constant term used in a calculation.

d. Correct. A numeric constant, or literal numeric constant, is a specific numeric value.

POINTS: 1

REFERENCES: 39
Understanding Data Types

QUESTION TYPE: Multiple Choice

HAS VARIABLES: False

LEARNING OBJECTIVES: PL&D.FARR.18.02.01 - Declare and use variables and constants

DATE CREATED: 9/12/2016 12:40 PM

DATE MODIFIED: 11/15/2019 4:11 PM

14. In most programming languages, before you can use any variable, you must include a ____ for it.

a. declaration b. definition
c. header d. proclamation

ANSWER: a

FEEDBACK:

a. Correct. A declaration is a statement that provides the data type, an identifier, and optionally an initial value.

b. Incorrect. The term *definition* describes assigning memory for a variable.

c. Incorrect. A header is the first line in a module (or method), which includes the module identifier and other necessary information.

d. Incorrect. *Proclamation* is not a commonly-used programming term for the statement that defines a variable.

POINTS: 1

REFERENCES: 41
Working with Variables

QUESTION TYPE: Multiple Choice

HAS VARIABLES: False
LEARNING OBJECTIVES: PL&D.FARR.18.02.01 - Declare and use variables and constants
DATE CREATED: 9/12/2016 12:40 PM
DATE MODIFIED: 11/15/2019 4:11 PM

15. The process of naming program variables and assigning a type to them is called ____ variables.

a. initializing b. declaring
c. identifying d. proclaiming

ANSWER: b
FEEDBACK:

a. Incorrect. Initializing a variable gives a variable its initial value.
b. Correct. Declaring a variable provides the data type, an identifier, and optionally an initial value.
c. Incorrect. *Identifying* is not a defined term for the naming and assignment of variables.
d. Incorrect. *Proclaiming* is not a defined term for the naming and assignment of variables.

POINTS: 1
REFERENCES: 40
Working with Variables
QUESTION TYPE: Multiple Choice
HAS VARIABLES: False
LEARNING OBJECTIVES: PL&D.FARR.18.02.01 - Declare and use variables and constants
DATE CREATED: 9/12/2016 12:40 PM
DATE MODIFIED: 11/15/2019 4:11 PM

16. Declaring a starting value for a variable is known as ____ the variable.

a. initializing b. declaring
c. defining d. identifying

ANSWER: a
FEEDBACK:

a. Correct. Initializing a variable gives a variable its initial value.
b. Incorrect. Declaring a variable provides the data type, an identifier, and optionally an initial value.
c. Incorrect. *Defining* describes when memory is assigned for a variable.
d. Incorrect. *Identifying* is not a defined term for assigning a starting value to a variable.

POINTS: 1
REFERENCES: 45
Assigning Values to Variables
QUESTION TYPE: Multiple Choice
HAS VARIABLES: False
LEARNING OBJECTIVES: PL&D.FARR.18.02.01 - Declare and use variables and constants
DATE CREATED: 9/12/2016 12:40 PM
DATE MODIFIED: 11/15/2019 4:11 PM

17. A variable’s unknown value is commonly called ____.

a. initial	b. default
c. random	d. garbage

ANSWER: d

FEEDBACK:

a. Incorrect. In most languages, a variable has an unknown value when it is declared but not initialized; this value is known as *garbage*.

b. Incorrect. Some programming languages assign a default value to an uninitialized variable; if they do, then the value is known.

c. Incorrect. A variable's unknown value when it has been declared but not initialized is known as *garbage*.

d. Correct. Although some languages use a default value for some variables, a variable's unknown value commonly is called *garbage*.

POINTS: 1

REFERENCES: 46
Assigning Values to Variables

QUESTION TYPE: Multiple Choice

HAS VARIABLES: False

LEARNING OBJECTIVES: PL&D.FARR.18.02.01 - Declare and use variables and constants

DATE CREATED: 9/12/2016 12:40 PM

DATE MODIFIED: 11/15/2019 4:11 PM

18. 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

ANSWER: c

FEEDBACK:

a. Incorrect. Hungarian notation is a form of camel casing in which a variable's data type is part of the identifier.

b. Incorrect. Pascal casing is a convention in which the first letter of a variable name is uppercase. It is sometimes called upper camel casing to distinguish it from lower camel casing.

c. Correct. Camel casing is the convention in which the variable starts with a lowercase letter and any subsequent word begins with an uppercase letter.

d. Incorrect. Turing notation is not a defined convention for naming variables.

POINTS: 1

REFERENCES: 42
Understanding a Declaration’s Identifier

QUESTION TYPE: Multiple Choice

HAS VARIABLES: False

LEARNING OBJECTIVES: PL&D.FARR.18.02.01 - Declare and use variables and constants

DATE CREATED: 9/12/2016 12:40 PM

DATE MODIFIED: 11/15/2019 4:11 PM

19. 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

ANSWER: b

FEEDBACK:

a. Incorrect. Hungarian notation is a form of camel casing in which a variable's data type is part of the identifier.

b. Correct. Pascal casing is a convention in which the first letter of a variable name is uppercase. It is sometimes called upper camel casing to distinguish it from lower camel casing.

c. Incorrect. Camel casing is the convention in which the variable starts with a lowercase letter and any subsequent word begins with an uppercase letter.

d. Incorrect. Turing notation is not a defined convention for naming variables.

POINTS: 1

REFERENCES: 43
Understanding a Declaration’s Identifier

QUESTION TYPE: Multiple Choice

HAS VARIABLES: False

LEARNING OBJECTIVES: PL&D.FARR.18.02.01 - Declare and use variables and constants

DATE CREATED: 9/12/2016 12:40 PM

DATE MODIFIED: 11/15/2019 4:11 PM

20. _____ 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

ANSWER: a

FEEDBACK:

a. Correct. Hungarian notation is a form of camel casing in which a variable's data type is part of the identifier.

b. Incorrect. Pascal casing is a convention in which the first letter of a variable name is uppercase. It is sometimes called upper camel casing to distinguish it from lower camel casing.

c. Incorrect. Turing notation is not a defined convention for naming variables.

d. Incorrect. Camel casing is the convention in which the variable starts with a lowercase letter and any subsequent word begins with an uppercase letter.

POINTS: 1

REFERENCES: 43
Understanding a Declaration’s Identifier

QUESTION TYPE: Multiple Choice

HAS VARIABLES: False

LEARNING OBJECTIVES: PL&D.FARR.18.02.01 - Declare and use variables and constants

DATE CREATED: 9/12/2016 12:40 PM

DATE MODIFIED: 11/15/2019 4:11 PM

21. The assignment operator is the _____ sign.

a. *   b. +

c. =   d. /

ANSWER: c

FEEDBACK:

a. Incorrect. The asterisk is used to multiply numerical values.

b. Incorrect. The plus sign is used to add numerical values.

c. Correct. The equal sign is the assignment operator.

d. Incorrect. The forward slash is used to divide numerical values.

POINTS: 1

REFERENCES: 45
Assigning Values to Variables

QUESTION TYPE: Multiple Choice

HAS VARIABLES: False

LEARNING OBJECTIVES: PL&D.FARR.18.02.01 - Declare and use variables and constants

DATE CREATED: 9/12/2016 12:40 PM

DATE MODIFIED: 11/15/2019 4:11 PM

22. 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

ANSWER: c

FEEDBACK:

a. Incorrect. Unnamed constants do not have identifiers like variables do.

b. Incorrect. A literal is an unnamed constant

c. Correct. A named constant can be assigned a value only once. Use a named constant when you want to assign a useful name for a value that will never be changed during a program's execution.

d. Incorrect. Constants may be named or unnamed.

POINTS: 1

REFERENCES: 46
Declaring Named Constants

QUESTION TYPE: Multiple Choice

HAS VARIABLES: False

LEARNING OBJECTIVES: PL&D.FARR.18.02.01 - Declare and use variables and constants

DATE CREATED: 9/12/2016 12:40 PM

DATE MODIFIED: 11/15/2019 4:11 PM

23. 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

ANSWER: a

FEEDBACK:

a. Correct. The rules of precedence, also called the order of operations, dictate that expressions within parentheses are valuated first, starting with the innermost parentheses if there are multiple sets; then multiplication and division from left to right; and then addition and subtraction from left to right. The assignment operator has a very low precedence, so the arithmetic operations are performed before the final assignment.

b. Incorrect. *Statement rules* is not a defined term. Every operator follows rules of precedence that dictate which operations in the same statement are carried out.

c. Incorrect. *Operator rules* is not a defined term. Every operator follows rules of precedence that dictate which operations in the same statement are carried out.

d. Incorrect. *Rules of arithmetic* is not a defined term. Every operator follows rules of precedence that dictate which operations in the same statement are carried out.

POINTS: 1

REFERENCES: 48
Performing Arithmetic Operations

QUESTION TYPE: Multiple Choice

HAS VARIABLES: False

LEARNING OBJECTIVES: PL&D.FARR.18.02.02 - Perform arithmetic operations

DATE CREATED: 9/12/2016 12:40 PM

DATE MODIFIED: 11/15/2019 4:11 PM

24. Depending on the programming language being used, modules are also known as ____ .

a. subroutines, procedures, or methods
b. subroutines, receptacles, or methods
c. tasks, functions, or methods
d. procedures, functions, or containers

ANSWER: a

FEEDBACK:

a. Correct. Programmers often refer to modules as *subroutines*, *procedures*, *functions*, or *methods*; the name usually reflects the programming language being used.

b. Incorrect. *Subroutine* and *method* are defined terms, but *receptacle* is not a defined term used to refer to a module.

c. Incorrect. *Function* and *method* are defined terms, but *task* is not a defined term used to refer to a module.

d. Incorrect. *Procedure* and *function* are defined terms, but *container* is not a defined term used to refer to a module.

POINTS: 1

REFERENCES: 51
Understanding the Advantages of Modularization

QUESTION TYPE: Multiple Choice

HAS VARIABLES: False

LEARNING OBJECTIVES: PL&D.FARR.18.02.03 - Appreciate the advantages of modularization

DATE CREATED: 9/12/2016 12:40 PM

DATE MODIFIED: 11/15/2019 4:11 PM

25. The process of breaking down a large program into modules is called ____.

a. decomposing
b. modularization
c. fragmentation
d. caching

ANSWER: b

FEEDBACK:

a. Incorrect. The process of breaking down a large program into modules is called *modularization*, although computer scientists also call it *functional decomposition*.

b. Correct. The process of breaking down a large program into modules is called *modularization*.

c. Incorrect. *Fragmentation* is not a defined term regarding breaking down a large program into modules.

d. Incorrect. *Caching* is not a defined term regarding breaking down a large program into modules. It is likely a confusion of storing data in CPU cache.

POINTS:	1
REFERENCES:	52 Understanding the Advantages of Modularization
QUESTION TYPE:	Multiple Choice
HAS VARIABLES:	False
LEARNING OBJECTIVES:	PL&D.FARR.18.02.03 - Appreciate the advantages of modularization
DATE CREATED:	9/12/2016 12:40 PM
DATE MODIFIED:	11/15/2019 4:11 PM

26. ____ is the process of paying attention to important properties while ignoring nonessential details.

a. Abstraction b. Modularization

c. Abbreviation d. Decomposition

ANSWER: a

FEEDBACK:

a. Correct. Abstraction is the process of paying attention to important properties while ignoring nonessential details.

b. Incorrect. The process of breaking down a large program into modules is called *modularization*.

c. Incorrect. *Abbreviation* is not a defined term regarding modularization.

d. Incorrect. The process of breaking down a large program into modules is called *modularization*, although computer scientists also call it *functional decomposition*.

POINTS:	1
REFERENCES:	52 Modularization Provides Abstraction
QUESTION TYPE:	Multiple Choice
HAS VARIABLES:	False
LEARNING OBJECTIVES:	PL&D.FARR.18.02.03 - Appreciate the advantages of modularization
DATE CREATED:	9/12/2016 12:40 PM
DATE MODIFIED:	11/15/2019 4:11 PM

27. Programmers say the statements that are contained in a module have been ____.

a. embedded b. decomposed

c. encapsulated d. modularized

ANSWER: c

FEEDBACK:

a. Incorrect. *Embedded* is not a defined term regarding modularization.

b. Incorrect. *Decomposed* is not a defined term regarding modularization. It is likely a confusion of the term *functional decomposition.*

c. Correct. Encapsulation is the act of containing instructions in a module.

d. Incorrect. *Modularized* describes a program that has been broken down into modules.

POINTS: 1

REFERENCES: 58
Modularizing a Program

QUESTION TYPE: Multiple Choice

HAS VARIABLES: False

LEARNING OBJECTIVES: PL&D.FARR.18.02.04 - Modularize a program

DATE CREATED: 9/12/2016 12:40 PM

DATE MODIFIED: 11/15/2019 4:11 PM

28. 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

ANSWER: c

FEEDBACK:

a. Incorrect. *Abstracted* is not a defined term.
b. Incorrect. A variable that is out of scope describes a variable existing in a different module.
c. Correct. Programmers say data is visible or in scope only within the module in which they are declared.
d. Incorrect. *In reference* is not a defined term.

POINTS: 1

REFERENCES: 59
Declaring Variables and Constants within Modules

QUESTION TYPE: Multiple Choice

HAS VARIABLES: False

LEARNING OBJECTIVES: PL&D.FARR.18.02.04 - Modularize a program

DATE CREATED: 9/12/2016 12:40 PM

DATE MODIFIED: 11/15/2019 4:11 PM

29. ____ variables and constants are known to the entire program.

a. Local b. Transient
c. Heap d. Global

ANSWER: d

FEEDBACK:

a. Incorrect. Local variable may only be accessed within the module they are declared.
b. Incorrect. *Transient* is not a defined term regarding variables.
c. Incorrect. *Heap* is not a defined term regarding variables.
d. Correct. Global variables and constants are known to the entire program.

POINTS: 1

REFERENCES: 60
Declaring Variables and Constants within Modules

QUESTION TYPE: Multiple Choice

HAS VARIABLES: False

LEARNING OBJECTIVES: PL&D.FARR.18.02.04 - Modularize a program

DATE CREATED: 9/12/2016 12:40 PM

DATE MODIFIED: 11/15/2019 4:11 PM

30. The mainline logic of almost every procedural computer program consists of these three distinct parts: _____ .

 a. housekeeping tasks, main 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

ANSWER: c

FEEDBACK:

 a. Incorrect. Main tasks are not defined as a part of the mainline logic of a procedural program.
 b. Incorrect. Clearing tasks are not defined as a part of the mainline logic of a procedural program.
 c. Correct. The mainline logic of a procedural program begins with housekeeping tasks, goes to detail loop tasks, and finishes with end-of-job tasks.
 d. Incorrect Math tasks are not defined as a part of the mainline logic of a procedural program.

POINTS: 1

REFERENCES: 60-61
Understanding the Most Common Configuration for Mainline Logic

QUESTION TYPE: Multiple Choice

HAS VARIABLES: False

LEARNING OBJECTIVES: PL&D.FARR.18.02.04 - Modularize a program

DATE CREATED: 9/12/2016 12:40 PM

DATE MODIFIED: 11/15/2019 4:11 PM

31. 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. flowchart d. data diagram

ANSWER: a

FEEDBACK:

 a. Correct. A hierarchy chart tells which modules exist within a program and which modules call others.
 b. Incorrect. A tree chart is not a specific type of chart but describes the structure of charts.
 c. Incorrect. A flow chart tells what tasks are to be performed within a module, when the modules are called, how a module executes, and why they are called.
 d. Incorrect. *Data diagram* is not a defined term and is likely a confusion of a diagram that shows a data structure.

POINTS: 1

REFERENCES: 64
Creating Hierarchy Charts

QUESTION TYPE: Multiple Choice

HAS VARIABLES: False

LEARNING OBJECTIVES: PL&D.FARR.18.02.05 - Create hierarchy charts

DATE CREATED: 9/12/2016 12:40 PM

DATE MODIFIED: 11/15/2019 4:11 PM

32. 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

ANSWER: d

FEEDBACK:

a. Incorrect. Large and complex programs require more planning and design than simple programs.

b. Incorrect. *Efficiency* is a subjective term surrounding the planning and design process.

c. Incorrect. Large and complex programs require more planning and design than simple programs.

d. Correct. Large and complex programs require more planning and design than simple programs.

POINTS: 1

REFERENCES: 66
Features of Good Program Design

QUESTION TYPE: Multiple Choice

HAS VARIABLES: False

LEARNING OBJECTIVES: PL&D.FARR.18.02.06 - Describe the features of good program design

DATE CREATED: 9/12/2016 12:40 PM

DATE MODIFIED: 11/15/2019 4:11 PM

33. In a flowchart, 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

ANSWER: b

FEEDBACK:

a. Incorrect. There is no defined flowchart symbol used to show an abstraction.

b. Correct. An annotation symbol is most often represented by a three-sided box connected by a dashed line to the step it references. Annotation symbols are used to hold comments or statements too long to fit neatly into a flowchart symbol.

c. Incorrect. There is no defined flowchart symbol used in abbreviation.

d. Incorrect. There is no defined flowchart symbol used in enumeration.

POINTS: 1

REFERENCES: 67
Using Program Comments

QUESTION TYPE: Multiple Choice

HAS VARIABLES: False

LEARNING OBJECTIVES: PL&D.FARR.18.02.06 - Describe the features of good program design

DATE CREATED: 9/12/2016 12:40 PM

DATE MODIFIED: 11/15/2019 4:11 PM

34. Programmers refer to programs that contain meaningful names as ____.

a. undocumented

b. procedurally documented

c. formally documented

d. self-documenting

ANSWER: d

FEEDBACK:

a. Incorrect. *Undocumented* refers to a program containing no documentation.

b. Incorrect. *Procedurally documented* is not a defined term.

c. Incorrect. *Formally documented* is not a defined term.

d. Correct. Self-documenting code explains itself to readers as a result of meaningful program names.

POINTS: 1

REFERENCES: 69
Choosing Identifiers

QUESTION TYPE: Multiple Choice

HAS VARIABLES: False

LEARNING OBJECTIVES: PL&D.FARR.18.02.06 - Describe the features of good program design

DATE CREATED: 9/12/2016 12:40 PM

DATE MODIFIED: 11/15/2019 4:11 PM

35. 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

ANSWER: c

FEEDBACK:

a. Incorrect. *Programming variable* is not a defined term. It likely describes a variable used in a program.

b. Incorrect. *Throw away variable* is not a defined term.

c. Correct. A temporary or work variable is used during a program's execution.

d. Incorrect. *Calculating variable* is not a defined term. It likely describes the use of a temporary variable used to perform calculations.

POINTS: 1

REFERENCES: 71
Designing Clear Statements

QUESTION TYPE: Multiple Choice

HAS VARIABLES: False

LEARNING OBJECTIVES: PL&D.FARR.18.02.06 - Describe the features of good program design

DATE CREATED: 9/12/2016 12:40 PM

DATE MODIFIED: 11/15/2019 4:11 PM

36. Whole number variables are known as ____________________ variables.

ANSWER: integer

POINTS: 1

REFERENCES: 39
Understanding Data Types

QUESTION TYPE: Completion

HAS VARIABLES: False
STUDENT ENTRY MODE: Basic
LEARNING OBJECTIVES: PL&D.FARR.18.02.01 - Declare and use variables and constants
DATE CREATED: 9/12/2016 12:40 PM
DATE MODIFIED: 11/15/2019 4:11 PM

37. Declaring a starting value is known as ____________________ the variable.
ANSWER: initializing
POINTS: 1
REFERENCES: 41
Working with Variables
QUESTION TYPE: Completion
HAS VARIABLES: False
STUDENT ENTRY MODE: Basic
LEARNING OBJECTIVES: PL&D.FARR.18.02.01 - Declare and use variables and constants
DATE CREATED: 9/12/2016 12:40 PM
DATE MODIFIED: 11/15/2019 4:11 PM

38. Each programming language has a few reserved ____________________ that are not allowed as variable names because they are part of the language’s syntax.
ANSWER: keywords
key words
POINTS: 1
REFERENCES: 42
Understanding a Declaration’s Identifier
QUESTION TYPE: Completion
HAS VARIABLES: False
STUDENT ENTRY MODE: Basic
LEARNING OBJECTIVES: PL&D.FARR.18.02.01 - Declare and use variables and constants
DATE CREATED: 9/12/2016 12:40 PM
DATE MODIFIED: 11/15/2019 4:11 PM

39. ____________________ tasks include any steps you must perform at the beginning of a program to get ready for the rest of the program.
ANSWER: Housekeeping
House-keeping
House keeping
housekeeping
house-keeping
house keeping
POINTS: 1
REFERENCES: 60
Understanding the Most Common Configuration for Mainline Logic
QUESTION TYPE: Completion
HAS VARIABLES: False

STUDENT ENTRY MODE: Basic
LEARNING OBJECTIVES: PL&D.FARR.18.02.04 - Modularize a program
DATE CREATED: 9/12/2016 12:40 PM
DATE MODIFIED: 11/15/2019 4:11 PM

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

ANSWER: comments
POINTS: 1
REFERENCES: 67
Using Program Comments
QUESTION TYPE: Completion
HAS VARIABLES: False
STUDENT ENTRY MODE: Basic
LEARNING OBJECTIVES: PL&D.FARR.18.02.06 - Describe the features of good program design
DATE CREATED: 9/12/2016 12:40 PM
DATE MODIFIED: 11/15/2019 4:11 PM

41. What does a data item’s data type describe?

ANSWER: 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
POINTS: 1
REFERENCES: 39
Understanding Data Types
QUESTION TYPE: Subjective Short Answer
HAS VARIABLES: False
STUDENT ENTRY MODE: Basic
LEARNING OBJECTIVES: PL&D.FARR.18.02.01 - Declare and use variables and constants
TOPICS: Critical Thinking
DATE CREATED: 9/12/2016 12:40 PM
DATE MODIFIED: 11/15/2019 4:11 PM

42. List three reasons for modularizing a large program.

ANSWER: 1) Modularization provides abstraction.
2) Modularization allows multiple programmers to work on a problem.
3) Modularization allows you to reuse your work more easily.
POINTS: 1
REFERENCES: 52
Understanding the Advantages of Modularization
QUESTION TYPE: Subjective Short Answer
HAS VARIABLES: False
STUDENT ENTRY MODE: Basic

LEARNING OBJECTIVES: PL&D.FARR.18.02.03 - Appreciate the advantages of modularization
TOPICS: Critical Thinking
DATE CREATED: 9/12/2016 12:40 PM
DATE MODIFIED: 11/15/2019 4:11 PM

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

ANSWER: 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.

POINTS: 1
REFERENCES: 54
Modularizing a Program
QUESTION TYPE: Subjective Short Answer
HAS VARIABLES: False
STUDENT ENTRY MODE: Basic
LEARNING OBJECTIVES: PL&D.FARR.18.02.04 - Modularize a program
TOPICS: Critical Thinking
DATE CREATED: 9/12/2016 12:40 PM
DATE MODIFIED: 11/15/2019 4:11 PM

44. Explain the purpose of detail loop tasks.

ANSWER: 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.

POINTS: 1
REFERENCES: 61
Understanding the Most Common Configuration for Mainline Logic
QUESTION TYPE: Subjective Short Answer
HAS VARIABLES: False
STUDENT ENTRY MODE: Basic
LEARNING OBJECTIVES: PL&D.FARR.18.02.04 - Modularize a program
TOPICS: Critical Thinking
DATE CREATED: 9/12/2016 12:40 PM
DATE MODIFIED: 11/15/2019 4:11 PM

45. What are end-of-job tasks?

ANSWER: 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.

POINTS: 1
REFERENCES: 61

Understanding the Most Common Configuration for Mainline Logic

QUESTION TYPE: Subjective Short Answer

HAS VARIABLES: False

STUDENT ENTRY MODE: Basic

LEARNING OBJECTIVES: PL&D.FARR.18.02.04 - Modularize a program

TOPICS: Critical Thinking

DATE CREATED: 9/12/2016 12:40 PM

DATE MODIFIED: 11/15/2019 4:11 PM

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

ANSWER: 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.

POINTS: 1

REFERENCES: 67
Features of Good Program Design

QUESTION TYPE: Subjective Short Answer

HAS VARIABLES: False

STUDENT ENTRY MODE: Basic

LEARNING OBJECTIVES: PL&D.FARR.18.02.06 - Describe the features of good program design

TOPICS: Critical Thinking

DATE CREATED: 9/12/2016 12:40 PM

DATE MODIFIED: 11/15/2019 4:11 PM

47. Explain the purpose of annotation symbols.

ANSWER: 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.

POINTS: 1

REFERENCES: 67
Using Program Comments

QUESTION TYPE: Subjective Short Answer

HAS VARIABLES: False

STUDENT ENTRY MODE: Basic

LEARNING OBJECTIVES: PL&D.FARR.18.02.06 - Describe the features of good program design

TOPICS: Critical Thinking

DATE CREATED: 9/12/2016 12:40 PM

DATE MODIFIED: 11/15/2019 4:11 PM

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

ANSWER: 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.

POINTS: 1

REFERENCES: 69
Choosing Identifiers

QUESTION TYPE: Subjective Short Answer

HAS VARIABLES: False

STUDENT ENTRY MODE: Basic

LEARNING OBJECTIVES: PL&D.FARR.18.02.06 - Describe the features of good program design

TOPICS: Critical Thinking

DATE CREATED: 9/12/2016 12:40 PM

DATE MODIFIED: 11/15/2019 4:11 PM

49. Explain the purpose of temporary variables.

ANSWER: 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.

POINTS: 1

REFERENCES: 71
Using Temporary Variables to Clarify Long Statements

QUESTION TYPE: Subjective Short Answer

HAS VARIABLES: False

STUDENT ENTRY MODE: Basic

LEARNING OBJECTIVES: PL&D.FARR.18.02.06 - Describe the features of good program design

TOPICS: Critical Thinking

DATE CREATED: 9/12/2016 12:40 PM

DATE MODIFIED: 11/15/2019 4:11 PM

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

ANSWER: 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.

POINTS:	1
REFERENCES:	74 Maintaining Good Programming Habits
QUESTION TYPE:	Subjective Short Answer
HAS VARIABLES:	False
STUDENT ENTRY MODE:	Basic
LEARNING OBJECTIVES:	PL&D.FARR.18.02.06 - Describe the features of good program design
TOPICS:	Critical Thinking
DATE CREATED:	9/12/2016 12:40 PM
DATE MODIFIED:	11/15/2019 4:11 PM

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

REFERENCES:	40, Working with Variables 41, Working with Variables, Understanding a Declaration’s Data Type 42, Understanding a Declaration’s Identifier 54, Modularization Allows You to Reuse Work 70, Choosing Identifiers 72, Writing Clear Prompts and Echoing Input 73, Writing Clear Prompts and Echoing Input
QUESTION TYPE:	Matching
HAS VARIABLES:	False
LEARNING OBJECTIVES:	PL&D.FARR.18.02.01 - Declare and use variables and constants PL&D.FARR.18.02.03 - Appreciate the advantages of modularization PL&D.FARR.18.02.06 - Describe the features of good program design
DATE CREATED:	9/12/2016 12:40 PM
DATE MODIFIED:	11/15/2019 4:11 PM

51. Named memory locations whose contents can vary or differ over time

ANSWER: g

POINTS: 1

52. A statement that provides a data type and an identifier for a variable

ANSWER: b

POINTS: 1

53. A program component’s name
ANSWER: e
POINTS: 1

54. Can hold digits and have mathematical operations performed on it
ANSWER: i
POINTS: 1

55. Can hold text, such as letters of the alphabet, and other special characters, such as punctuation marks
ANSWER: d
POINTS: 1

56. The feature of programming languages that prevents assigning values of an incorrect data type
ANSWER: j
POINTS: 1

57. The feature of programs that assures you a module has been tested and proven to function correctly
ANSWER: a
POINTS: 1

58. A list of every variable name used in a program, along with its type, size, and description
ANSWER: h
POINTS: 1

59. A message that is displayed on a monitor to ask the user for a response and perhaps explain how that response should be formatted
ANSWER: f
POINTS: 1

60. The act of repeating input back to a user either in a subsequent prompt or in output
ANSWER: c
POINTS: 1