Chapter Review and Exercises

Key Terms

American Standard Code for Information Interchange
ASCII code
assignment operator
Case (Switch) statement
character
comparison operator
defensive programming
dual-alternative structure
`Else` clause
error trap
`If-Then` structure
`If-Then-Else` structure
logical operator
menu-driven (program)
multiple-alternative structure
relational operator
selection (decision) control structure
single-alternative structure
test condition
`Then` clause
truth tables

Chapter Summary

In this chapter, we have discussed the following topics:

1. The single-alternative selection structure as follows:
   - It contains a single block of statements to be either executed or skipped
   - It is implemented by an `If-Then` statement
2. The dual-alternative selection structure as follows:
   - It contains two blocks of statements, one of which is to be executed while the other is to be skipped
   - It is implemented by an `If-Then-Else` statement
3. The multiple-alternative selection structure as follows:
   - It contains more than two blocks of statements, one of which is to be executed while the others are to be skipped
   - It can be implemented by a sequence of `If-Then` statements, nested `If-Then-Else` statements, or a `Case` statement
4. Relational and logical operators as follows:
   - Relational operators are the comparison (equal) operator, which compares the value of one thing to another (`==`); the not equal operator (`!=`); the less than operator (`<`); the less than or equal to operator (`<=`); the greater than operator (`>`); and the greater than or equal to operator (`>=`)
   - Logical operators are `NOT`, `AND`, and `OR`
   - The order of operations in the absence of parentheses is that arithmetic operations are done first (in their usual order); then relational operations (all with equal precedence so they can be done in any order); and finally, the logical operators, `NOT`, `AND`, and `OR`, executed in that order

5. The ASCII code as follows:
   - In ASCII code, a number is associated with each character
   - ASCII code can be used to order the set of characters
   - ASCII code can be used to order arbitrary character strings
6. Defensive programming as follows:
   - Programmers must anticipate and prevent errors that might result from using improper data
   - Division by zero can cause a program crash; a selection structure can be used to prevent instances of division by zero
   - The square root function, `Sqrt()`, is a commonly used function but attempting to take the square root of a negative number can cause a program crash; defensive programming ensures that a program will not be asked to take the square root of a negative number
7. The built-in square root function as follows:
   - It is of the form `Sqrt(X)`, where `X` is a number, numeric variable, or arithmetic expression
   - It may be used anywhere in a program where a numeric constant is valid
8. Menu-driven programs as follows:
   - They present options for the user in the form of menus
   - They use a multiple-alternative structure to handle the user's option selection

Review Exercises

Fill in the Blank

1. A single-alternative structure is also known as a(n) _____ structure.
2. A dual-alternative structure is also known as a(n) _____ structure.

For Exercises 3–8: Replace the blank with one of the following words: arithmetic relational, logical

3. `<=` is a(n) _____ operator
4. `+` is a(n) _____ operator
5. `%` is a(n) _____ operator
6. `OR` is a(n) _____ operator
7. `NOT` is a(n) _____ operator
8. `!=` is a(n) _____ operator

Multiple Choice

9. Which expression is equivalent to the following expression but does not use the `NOT` operator: `NOT (A > B)`

   a. `A < B` b. `A <= B` c. `B < A` d. `B <= A`

10. Which expression is equivalent to the following expression:
    `A > 8 AND A < 18`

    a. `NOT (A < 8) AND NOT (A > 18)`

b. `NOT (A <= 8) AND NOT (A >= 18)`
c. `NOT (A > 8 OR A < 18)`
d. `A < 8 OR A > 18`

11. A multiple-alternative structure cannot be implemented by using which of the following:
    a. A single `If-Then` statement
    b. Several `If-Then` statements
    c. Several `If-Then-Else` statements
    d. A single `Case` statement

12. If `Char1 = "/"` and `Char2 = "?"`, which of the following expressions are true?
    a. `Char1 < Char2`
    b. `Char1 <= Char2`
    c. `Char1 > Char2`
    d. `Char1 >= Char2`

13. The term defensive programming refers to which of the following:
    a. Ensuring that input data are in the proper range
    b. Ensuring that a division by 0 does not take place
    c. Ensuring that the square root operation is valid
    d. Techniques that include all the above points

True or False

14. If `X = 0`, determine whether each of the following expressions is true or false.
    a. T F `X >= 0`
    b. T F `2 * X + 1 != 1`

15. If `First = "Ann"`, determine whether each of the following expressions is true or false.
    a. T F `First == "ann"`
    b. T F `First != "Ann"`
    c. T F `First < "Nan"`
    d. T F `First >= "Anne"`

16. Let `X = 1` and `Y = 2`. Then determine whether each of the following expressions is true or false.
    a. T F `X >= X OR Y >= X`
    b. T F `X > X AND Y > X`
    c. T F `X > Y OR X > 0 AND Y < 0`
    d. T F `NOT(NOT(X == 0) AND NOT (Y == 0))`

17. If `X = 0` and `Response = "Yes"`, determine whether each of the following expressions is true or false.
    a. T F `(X == 1) OR (Response == "Yes")`
    b. T F `(X == 1) AND (Response == "Yes")`
    c. T F `NOT (X == 0)`

18. If `Num1 = 1` and `Num2 = 2`, determine whether each of the following expressions is true or false.
    a. T F `(Num1 == 1) OR (Num2 == 2) AND (Num1 == Num2)`
    b. T F `((Num1 == 1) OR (Num2 == 2)) AND (Num1 == Num2)`
    c. T F `NOT (Num1 == 1) AND NOT (Num2 == 2)`
    d. T F `NOT (Num1 == 1) OR NOT (Num2 == 2)`

19. T F The ASCII coding scheme associates a number between 0 and 127 with every lowercase and uppercase letter, as well as many other characters.

20. T F If `Char1` and `Char2` are characters, then `Char1 == Char2` if and only if their ASCII codes are the same.

21. T F If `Name = "John"`, then `Name > " John"`.

22. T F If `Name = "John"`, then `Name >= "JOHN"`.

23. T F `"**?" < "***"`.

24. T F `"** " < "***"`.

25. T F A `Case` statement can be used to select an alternative based on the value of a variable of character type.

26. T F A menu-driven program requires the user to memorize a list of commands in order to select options offered by the program.

27. T F In a menu-driven program, the options on the main menu usually correspond to separate program modules.

Short Answer

28. Suppose that `X = 'A'`. What is displayed when code corresponding to the following program segment is run?

```
If X == 'B' Then
  Write "Hi"
End If
Write "Bye"
```

29. Suppose that `X = 0`. What is displayed when code corresponding to the following program segment is run?

```
If X == 1 Then
  Write "Hi"
Else
  Write "Why?"
End If
Write "Bye"
```

30. Give the ASCII code for each of the following characters:
    a. `&`
    b. `2`
    c. `@`
31. What character corresponds to each of the following ASCII codes?
    a. 33
    b. 65
    c. 126
32. If the string `"}123*"` is less than the string `S` (according to the ASCII code), with which character must `S` begin?
33. Give the ASCII code for each of the following words, character by character:
    a. why?
    b. Oh my!
34. Write a program segment that inputs `Age` and displays `"You are too young to vote"` if `Age` is less than 18 (and displays nothing else).
35. Draw a flowchart that corresponds to Exercise 34.
36. Write a program segment that inputs `Age`, displays `"Yes, you can vote"` if `Age` is 18 or older and displays `"You are too young to vote"` if `Age` is less than 18, and displays nothing else. Remember to use the comparison operator when checking the value of `Age` in your `If-Then-Else` statements.
37. Draw a flowchart that corresponds to Exercise 36.
38. Write a program segment that contains two `If-Then` statements and which inputs `Num`, then displays `"Yes"` if `Num == 1` and displays `"No"` otherwise.
39. Write a program segment that contains a single `If-Then-Else` statement and which inputs `Num`, then displays `"Yes"` if `Num == 1` and displays `"No"` otherwise.
40. List the programming symbols for the relational operators.
41. List three logical operators used in programming.
42. Write expressions equivalent to the following without using the `NOT` operator:
    a. `NOT (N > 0)`
    b. `NOT ((N >= 0) AND (N <= 5))`
43. Write expressions equivalent to the following using a single *relational* operator:
    a. `(X > 1) AND (X > 5)`
    b. `(X == 1) OR (X > 1)`

Exercises 44–48: Be sure to use the hierarchy of order of operations.

44. Evaluate the following expressions, using the values for A, B, and C as given. Note that T means true and F means false.

```
A = T          B = F          C = F          D = T
```

a. `A OR B OR C OR D`

b. `A AND B AND C AND D`

c. `A AND B OR C AND D`

d. `A OR B AND C OR D`

45. Evaluate the following expressions, using the values for J, K, L, and M as given. Note that T means true and F means false.

```
J = F          K = F          L = T          M = T
```

a. `NOT J OR K AND L OR M`

b. `NOT J AND NOT K AND NOT L AND NOT M`

c. `J AND K OR L AND M`

d. `NOT J OR NOT K OR NOT L OR NOT M`

46. Evaluate the following expressions, using the values for W, X, Y, and Z as given. Note that T means true and F means false.

```
W = T          X = T          Y = F          Z = T
```

a. `W OR X OR X AND Z`

b. `W OR X OR X AND NOT Z`

c. `W AND Y AND Y OR X AND Z`

d. `W AND X AND NOT Y AND Z`

47. Evaluate the following expressions, using the values for A, B, C, and D as given. Note that T means true and F means false.

```
A = T          B = T          C = T          D = F
```

a. `A OR B OR C OR D`

b. `A OR (B OR (C OR D))`

c. `A AND B AND C AND D`

d. `A AND B AND (C OR D)`

48. Evaluate the following expressions, using the values for R, S, P, U, and W as given. Note that T means true and F means false.

```
R = F          S = F          P = F          U = T          W = T
```

a. `(S OR P AND U) OR (NOT R AND NOT S)`

b. `(R AND NOT S) OR (P AND W) AND NOT (U OR W) OR S`

c. `(U AND W) OR (R AND S AND NOT P) OR NOT (U AND W)`

49. Evaluate the following expression, given that `X = 3`, `Y = 5`, and `Z = 2`. Your answer will either be true or false.

```
(NOT (X < Y) OR (Y > Z)) AND (X > Z)
```

50. Evaluate the following expression, given that `A = 3` and `B = 2`. Your answer will either be true or false.

```
(A + 6)^2 - (B + 4 * A) <= A^3 + B * 5
```

51. Evaluate the following expression, given that `A = 1` and `B = 3`. Your answer will either be true or false.

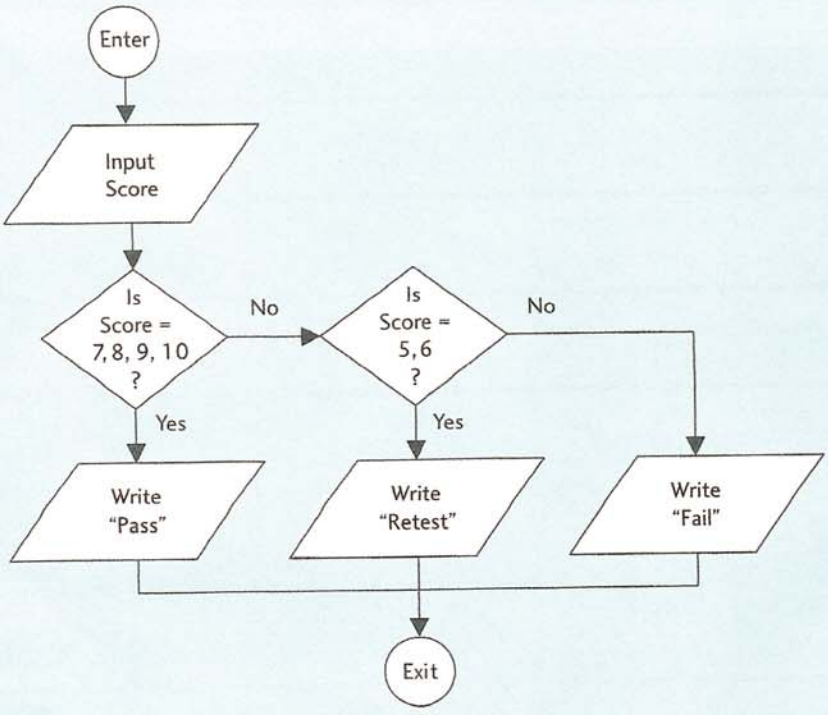
```
B^3 % A > B^3 * A
```

52. Write 3 short program segments that input `Num`; then display `"Yes"` if `Num == 1`, display `"No"` if `Num == 2`, and display `"Maybe"` if `Num == 3`. Implement this program segment by
    a. A sequence of `If-Then` statements.
    b. Nested `If-Then-Else` statements.
    c. A `Case` statement.

53. Write a program segment that inputs a number `X` and does all of the following:
    a. Displays the reciprocal of its square root, $1/\text{Sqrt}(X)$, if `X > 0`
    b. Displays `"Error: Division by zero"`, if `X = 0`
    c. Displays `"Error: Square root of negative number"`, if `X < 0`

54. Which type of selection structure does the flowchart in Figure 3.9 represent?

Figure 3.9 Flowchart for Exercises 54–57



55. Write pseudocode that uses `If-Then` statements to achieve the same action as that indicated by the flowchart shown in Figure 3.9. Assume that `Score` is an integer between 1 and 10 inclusively.

56. Write pseudocode that uses nested `If-Then-Else` statements to implement the flowchart in Figure 3.9. Assume that `Score` is an integer between 1 and 10 inclusive.

57. Write pseudocode that uses a `Case` statement to achieve the same action as that indicated by the flowchart in Figure 3.9. Assume that `Score` is an integer between 1 and 10 inclusively.

58. What is displayed when code corresponding to the following program segment is run?

```
Set X = 5
If X > 0 Then
  Write X
End If
If NOT ((X == 0) OR (X < 0)) Then
  Write "Not"
End If
If (X ^ 2 >= 0) AND ((2 * X - 1) != 0) Then
  Write "And"
End If
```

59. Suppose that we replace the first statement in Exercise 58 by

```
Set X = -5
```

What would be displayed when the code is run using this value?

60. Using the code shown below, what will be displayed when the input is as follows:

a. −1

b. 0

c. 1

```
Input Number
If Number < 0 Then
  Write "1"
Else
  If Number == 0 Then
    Write "2"
  Else
    Write "3"
  End If
End If
Write "DONE"
```

61. What is displayed when code corresponding to the following program segment is run and the input is as follows:

a. −1

b. 0

c. 1

```
Set Number1 = 1
Input Number2
Select Case Of Number2
  Case -1:
    Write "A"
    Break
  Case 0:
    Write "B"
    Break
  Case Number1:
    Write "C"
    Break
  End Case
```

62. This program segment is supposed to display `HELLO` if `Grade == "A"` and display `GOODBYE`, otherwise. Correct the logic error so that it works correctly.

```
If Grade != "A" Then
  Write "HELLO"
Else
  Write "GOODBYE"
End If
```

63. This program segment is supposed to display `NEGATIVE` if `Number` is less than `0`, `SMALL` if `Number` lies between `0` and `5` (inclusive), and `LARGE` if `Number` is greater than `5`. Correct the logic error so that this pseudocode works correctly.

```
If Number < 0 Then
  Write "NEGATIVE"
Else
  If Number > 5 Then
    Write "SMALL"
  Else
    Write "LARGE"
  End If
End If
```

64. What is displayed when code corresponding to the following program segment is run?

```
Set Y = 1
If Sqrt(Y-1) == 0 Then
  Write "YES"
Else
  Write "NO"
End If
```

65. What is displayed when code corresponding to the program segment below is run with the following:

a. `X = 4`

b. `X = 0`

66.

67.

Pro

All Pr

For ea to de techn

1.

2.

3.

4.

```
Declare X As Integer
Input X
If X != 0 Then
  Write 1/X
Else
  Write "The reciprocal of 0 is not defined."
End If
```

66. Which block of statements, the `Then` clause or the `Else` clause, provides the error trap in Exercise 65?

67. Consider the following statement:

    ```
    Set Number3 = Sqrt(Number1)/Number2
    ```

    If `Number1` and `Number2` were input by the user and properly validated to ensure that `Number1` is greater than or equal to `0` and `Number2` is not zero, is any additional defensive programming necessary?

Programming Problems

All Programming Problems in this section are RAPTOR friendly.

For each of the following Problems, use the top-down modular approach and pseudocode to design a suitable program to solve it. Where appropriate, use defensive programming techniques.

1. Input a number entered by the user and display `"Positive"` if it is greater than zero, `"Negative"` if it is less than zero, and `"Zero"` if it is equal to zero.
2. Develop a menu-driven program that inputs two numbers and, at the user's option, finds their sum, difference, product, or quotient.
3. Input a number (`X`) and find the area (`Area`) of each of the following:
   - A square with side `X`, `Area = X * X`
   - A circle with radius `X`, `Area = 3.14 * X^2`
   - An equilateral triangle with side `X`, `Area = Sqrt(3)/4 * X^2`

   Note: Because `X` represents a dimension, we require that `X > 0`. Be sure to include this requirement in your program.
4. Consider the equation $Ax^2 + B = 0$.
   - If `B/A < 0`, this equation has two solutions. The solutions are:
     1. X_1 = `Sqrt(-B/A)`
     2. X_2 = `-Sqrt(-B/A)`
   - If `B/A = 0`, this equation has one solution which is `X = 0`
   - If `B/A > 0`, this equation has no real number solutions

   Write a program to have the user input any numbers for the coefficients, A and B, for this equation. If `A = 0`, terminate the program. Otherwise, solve the equation.

5. Compute the income tax due on a taxable income entered by the user, given the data as shown in the following table. Be sure to include error checking to make sure the user does not enter a negative number.

Taxable Income		Tax Due
From	To	
$0	$50,000	$0 + 5% of amount over $0
$50,000	$100,000	$2,500 + 7% of amount over $50,000
$100,000	...	$6,000 + 9% of amount over $100,000

6. Write a program that allows the user to input a total dollar amount for an online shopping order and computes and outputs the shipping cost based on the following schedule:

Order Total	Ship within USA	Ship to Canada
Less than $50.00	$6.00	$8.00
$50.01–$100.00	$9.00	$12.00
$100.01–$150.00	$12.00	$15.00
Over $150.00	Free	Free

7. Write a program that allows the user to enter his or her name. The program should output a username for entry into a Web site. The user should be prompted to enter a first name, middle initial, and last name. If the user doesn't have a middle initial, the entry should be "none." In this case, the output will be a username that concatenates the first and last names with a dot between them. If the user has a middle initial, the output username should be in the form `first.middle_initial.last`. For example, a user whose name is Harold Nguyen would have `Harold.Nguyen` for his username and a user whose name is Maria Anna Lopez would have `Maria.A.Lopez` for a username.