

TEST BANK



**ORACLE 10g
DEVELOPER**

PL/SQL PROGRAMMING

JOAN CASTEL

Includes an Oracle 10g
Database Software

ch02

True/False

Indicate whether the statement is true or false.

- ___ 1. The term anonymous blocks refers to blocks of code that are not stored for reuse and do not exist after being executed.
- ___ 2. The only required sections of a PL/SQL block are DECLARE and END.
- ___ 3. The BEGIN section of a PL/SQL block contains code that creates variables, cursors, and types.
- ___ 4. Assignment statements are used to put or change the values of variables.
- ___ 5. The keyword DEFAULT can be used in place of the := symbol to assign initial values to the variables within the declaration statement.
- ___ 6. The LOOP statement is a mechanism that allows the checking of a condition to determine if statements should or should not be processed.
- ___ 7. When an IF statement checks only one condition and performs actions only if the condition is TRUE, it is referred to as a simple IF condition.
- ___ 8. With respect to processing efficiency, the less code that has to be processed, the faster the program runs.
- ___ 9. The syntax of the following code fragment is correct:
IF rec_order.state = 'VA' THEN
 lv_tax_num := rec_order.sub * .06;
ELSEIF rec_order.state = 'ME' THEN
 lv_tax_num := rec_order.sub * .05;
ELSE
 lv_tax_num := rec_order.sub * .04;
END IF
- ___ 10. The syntax of the following code fragment is correct.
IF rec_order.state = 'VA' THEN
 lv_tax_num := rec_order.sub * .06;
ELS IF rec_order.state = 'ME' THEN
 lv_tax_num := rec_order.sub * .05;
END IF
- ___ 11. The CASE statement begins with the keyword CASE followed by a selector that is typically a variable name.
- ___ 12. The following code fragment is a correct example of the use of a basic loop.
BEGIN
 LOOP
 DBMS_OUTPUT.PUT_LINE(lv_cnt_num);
 lv_cnt_num := lv_cnt_num + 1;

```
END LOOP;
END;
```

- ___ 13. An infinite loop causes a program to loop indefinitely, disrupting the ability of the code to continue with any processing beyond the loop.
- ___ 14. The EXIT WHEN clause ensures that a basic loop runs at least once.
- ___ 15. The following code fragment is a correct example of the use of a WHILE loop.
- ```
BEGIN
 WHILE lv_cnt_num <= 5
 DBMS_OUTPUT.PUT_LINE(lv_cnt_num);
 lv_cnt_num := lv_cnt_num + 1;
 END LOOP;
END;
```
- \_\_\_ 16. The following loop iterates four times.
- ```
BEGIN
  WHILE lv_cnt_num <= 5 LOOP
    DBMS_OUTPUT.PUT_LINE( lv_cnt_num );
    lv_cnt_num := lv_cnt_num + 1;
  END LOOP;
END;
```
- ___ 17. The following loop terminates when the lv_cnt_num variable holds a value of 6.
- ```
BEGIN
 WHILE lv_cnt_num <= 5 LOOP
 DBMS_OUTPUT.PUT_LINE(lv_cnt_num);
 lv_cnt_num := lv_cnt_num + 1;
 END LOOP;
END;
```
- \_\_\_ 18. The following code fragment is a correct example of the use of a FOR loop.
- ```
BEGIN
  FOR i IN 1..5
    DBMS_OUTPUT.PUT_LINE( i );
  END LOOP;
END;
```
- ___ 19. Even though the EXIT clause can be used in any type of loop, it is considered good form to use the EXIT clause only in basic loops.

Modified True/False

Indicate whether the statement is true or false. If false, change the identified word or phrase to make the statement true.

- ___ 20. The BEGIN section of a PL/SQL block contains code that creates variables, cursors, and types.
-

- ___ 21. A(n) scalar variable can hold only a single value. _____
- ___ 22. The DECLARE section of the PL/SQL block contains all the processing action, or programming logic.

- ___ 23. The EXCEPTION section of a PL/SQL block contains handlers that allow you to control what the application will do if an error occurs. _____
- ___ 24. Variables are named memory areas that hold values to allow retrieval and manipulation of values within your programs. _____
- ___ 25. The common data types used for cursor variables include character, numeric, date, and Boolean.

- ___ 26. When an IF statement checks only one condition and performs actions only if the condition is TRUE, it is referred to as a(n) simple IF condition. _____
- ___ 27. The Searched CASE statement does not use a selector but individually evaluates conditions that are placed in the WHERE clauses. _____
- ___ 28. A(n) CASE expression evaluates conditions and returns a value in an assignment statement.

- ___ 29. Loops are used for situations in which we need to repeat a line or lines of code within our block.

- ___ 30. The basic loop uses the LOOP and END LOOP markers to begin and end the loop code.

- ___ 31. If the WHERE clause is not included the basic loop, the result is the programmer's nightmare of the infinite loop. _____
- ___ 32. The basic loop dictates exactly how many times the loop should run in the opening LOOP clause.

- ___ 33. To keep code efficient and minimize statement processing, any statements that are dynamic in nature should be placed outside a loop. _____

Multiple Choice

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

- ___ 34. The only required sections of a PL/SQL block are the ___ sections.
a. BEGIN & DECLARE c. BEGIN & END
b. DECLARE & EXCEPTION d. EXCEPTION & END
- ___ 35. The ___ section of a PL/SQL block contains code that creates variables, cursors, and types.
a. DECLARE c. EXCEPTION
b. BEGIN d. END
- ___ 36. ___ are used to change the values of variables.

- a. Loops
- b. Assignment statements
- c. Exceptions
- d. Blocks

___ 37. The ___ section of a PL/SQL block contains handlers that allow you to control what the application will do if an error occurs when the executable statements are processed.

- a. EXCEPTION
- b. BEGIN
- c. DECLARE
- d. END

___ 38. ___ are named memory areas that hold values to allow the retrieval and manipulation of values in a program.

- a. Loops
- b. Assignment statements
- c. Blocks
- d. Variables

___ 39. Which of the following lines of code is syntactically correct?

a. DECLARE
 order NUMBER;
 departure DATE;
BEGIN
 ---- executable statements ---
END

b. DECLARE
 order NUMBER;
 departure DATE
BEGIN
 ---- executable statements ---
END

c. DECLARE
 order NUMBER(2);
 departure DATE;
BEGIN;
 ---- executable statements ---
END

d. DECLARE
 order NUMBER(3);
 departure DATE;
BEGIN
 ---- executable statements ---
END;

___ 40. Which of the following initializes the variable *order*?

a. DECLARE
 order NUMBER(2);
 departure DATE;
BEGIN
 ---- executable statements ---
END;

b. DECLARE
 order NUMBER(2) = 0;

```
    departure DATE;
BEGIN
    ---- executable statements ---
END;
```

c. DECLARE
 order NUMBER(2) =: 0;
 departure DATE;
BEGIN
 ---- executable statements ---
END;

d. DECLARE
 order NUMBER(2) := 0;
 departure DATE;
BEGIN
 ---- executable statements ---
END;

___ 41. Which of the following PL/SQL blocks requires the variable to always contain a particular value within the block?

a. DECLARE
 order NUMBER(2) := 0;
 departure DATE;
BEGIN
 ---- executable statements ---
END;

b. DECLARE
 order NUMBER(2,2) := .06;
 departure DATE;
BEGIN
 ---- executable statements ---
END;

c. DECLARE
 order CONSTANT NUMBER(2,2) := .02;
 departure DATE;
BEGIN
 ---- executable statements ---
END;

d. DECLARE
 order NUMBER(2) CONSTANT := .03;
 departure DATE;
BEGIN
 ---- executable statements ---
END;

___ 42. DECLARE
 order NUMBER(2) := 4;

```
total_amt NUMBER(2);
BEGIN
total_amt := order * 8;
END;
```

According to the statement block above, what value is stored in the variable *total_amt*?

- a. 4
- b. 8
- c. 12
- d. 32

___ 43. Which of the following statement blocks correctly uses a scalar variable in an assignment statement?

a. DECLARE
order NUMBER(2) := 4;
total_amt NUMBER(2);
BEGIN
total_amt = 12;
END;

b. DECLARE
order NUMBER(2) := 4;
total_amt NUMBER(2);
BEGIN
total_amt := 12 * order;
END;

c. DECLARE
order NUMBER(2) := 4;
total_amt NUMBER(2);
BEGIN
order := total_amt *12;
END;

d. DECLARE
order NUMBER(2) := 4;
total_amt NUMBER(2);
BEGIN
total_amt := total_amt *12;
END;

___ 44. The statements that are used to control the flow of logic processing in your programs are commonly referred to as ___.

- a. exceptions
- b. control structures
- c. pragma statements
- d. index-by tables

___ 45. Which of the following statements is correct?

a. IF order > 5
prize = 'yes';
END IF;

b. IF order > 5 THEN
prize = 'yes';
ENDIF

- c. IF order > 5 THEN;
 prize = 'yes';
END IF;
- d. IF order > 5 THEN
 prize := 'yes';
END IF;

_____ 46. Which of the following code fragments would not raise an error?

- a. IF rec_order.state = 'VA' THEN
 lv_tax_num := rec_order.sub * .06;
ELSEIF rec_order.state = 'ME' THEN
 lv_tax_num := rec_order.sub * .05;
ELSE
 lv_tax_num := rec_order.sub * .04;
END IF;
- b. IF rec_order.state = 'VA' THEN
 lv_tax_num := rec_order.sub * .06;
ELSE IF rec_order.state = 'ME' THEN
 lv_tax_num := rec_order.sub * .05;
ELSE
 lv_tax_num := rec_order.sub * .04;
END IF;
- c. IF rec_order.state = 'VA' THEN
 lv_tax_num := rec_order.sub * .06;
ELSIF rec_order.state = 'ME' THEN
 lv_tax_num := rec_order.sub * .05;
ELSE
 lv_tax_num := rec_order.sub * .04;
END IF;
- d. IF rec_order.state = 'VA' THEN
 lv_tax_num := rec_order.sub * .06;
ELS IF rec_order.state = 'ME' THEN
 lv_tax_num := rec_order.sub * .05;
ELSE
 lv_tax_num := rec_order.sub * .04;
END IF;

_____ 47. Which of the following code fragments would not raise an error?

- a. IF rec.state = 'VA' OR 'PA' THEN
 a := b * .06;
ELSE
 a := b * .04;
END IF;
- b. IF rec.state = 'VA' OR rec.state = 'PA' THEN
 a := b * .06;
ELSE


```
    a := b * .04;
END IF;
```

c. IF rec.state = 'VA' OR rec.state = 'PA'

```
    a := b * .06;
ELSE
    a := b * .04;
END IF;
```

d. IF rec.state = 'VA' OR rec.state = 'PA' THEN

```
    a := b * .06;
ELSE
    a := b * .04;
END IF
```

- ___ 48. Which of the following does not use a selector, but individually evaluates conditions that are placed in WHEN clauses?
- a. Control statements
 - b. Searched CASE
 - c. Loops
 - d. CASE expression
- ___ 49. Which of the following evaluates conditions and returns a value in an assignment statement?
- a. Searched CASE
 - b. Basic loop
 - c. CASE expression
 - d. Control statement
- ___ 50. Which of the following statements is true?
- a. The WHEN clause of a CASE statement ends with a semicolon.
 - b. The WHEN clause of a CASE statement ends with "END CASE;".
 - c. The WHEN clause of a CASE expression does not end with a semicolon.
 - d. The WHEN clause of a CASE statement ends with "ENDCASE".
- ___ 51. Which of the following allow us to repeat the processing of a desired portion of code?
- a. Functions
 - b. Looping constructs
 - c. IF statements
 - d. CASE expressions
- ___ 52. The ___ uses the LOOP and END LOOP markers to begin and end the loop code.
- a. basic loop
 - b. cursor
 - c. index-by table
 - d. general loop
- ___ 53. Which of the following code fragments would not raise an error?
- a. BEGIN
 LOOP
 DBMS_OUTPUT.PUT_LINE(lv_cnt_num);
 lv_cnt_num := lv_cnt_num + 1;
 END LOOP;
END;
 - b. BEGIN
 LOOP
 DBMS_OUTPUT.PUT_LINE(lv_cnt_num);
 EXIT WHEN lv_cnt_num >= 5;
 lv_cnt_num := lv_cnt_num + 1;

END;

c. BEGIN
 LOOP
 DBMS_OUTPUT.PUT_LINE(lv_cnt_num);
 EXIT WHEN lv_cnt_num >= 5;
 lv_cnt_num := lv_cnt_num + 1
 END LOOP
END

d. BEGIN
 LOOP
 DBMS_OUTPUT.PUT_LINE(lv_cnt_num);
 EXIT WHEN lv_cnt_num >= 5;
 lv_cnt_num := lv_cnt_num + 1;
 END LOOP;
END;

___ 54. If the EXIT WHEN clause is not included in a basic loop, then the result is a(n) ____.

a. infinite loop	c. RAISE_APPLICATION_ERROR
b. exception	d. SQLCODE error

___ 55. Which of the following clauses ensures that a basic loop runs at least once?

a. EXIT WHEN	c. CASE
b. WHERE	d. LOOP

___ 56. BEGIN
 LOOP
 DBMS_OUTPUT.PUT_LINE(lv_cnt_num);
 lv_cnt_num := lv_cnt_num + 1;
 EXIT WHEN lv_cnt_num >= 5;
 END LOOP;
END;

Which of the statements in the code fragment above ensures that the loop executes at least once?

- a. LOOP
- b. lv_cnt_num := lv_cnt_num + 1;
- c. EXIT WHEN lv_cnt_num >= 5;
- d. DBMS_OUTPUT.PUT_LINE(lv_cnt_num);

___ 57. Which of the following code fragments would not raise an error?

a. BEGIN
 WHILE lv_cnt_num <= 5
 DBMS_OUTPUT.PUT_LINE(lv_cnt_num);
 lv_cnt_num := lv_cnt_num + 1;
 END LOOP;
END;

b. BEGIN
 WHILE lv_cnt_num <= 5
 DBMS_OUTPUT.PUT_LINE(lv_cnt_num);
 lv_cnt_num := lv_cnt_num + 1;

END;

- c. BEGIN
 WHILE lv_cnt_num <= 5 LOOP
 DBMS_OUTPUT.PUT_LINE(lv_cnt_num);
 lv_cnt_num := lv_cnt_num + 1;
 END LOOP;
- d. BEGIN
 WHILE lv_cnt_num <= 5 LOOP
 DBMS_OUTPUT.PUT_LINE(lv_cnt_num);
 lv_cnt_num := lv_cnt_num + 1;
 END LOOP;
END;

____ 58. BEGIN
 WHILE lv_cnt_num <= 5 LOOP
 DBMS_OUTPUT.PUT_LINE(lv_cnt_num);
 lv_cnt_num := lv_cnt_num + 1;
 END LOOP;
END;

According to the code fragment above, how many times does the loop iterate?

- a. 3 c. 5
b. 4 d. 6

____ 59. FOR i IN 1..tbl_roast.COUNT LOOP
 lv_tot_num := lv_tot_num + tbl_roast(i);
END LOOP;

In the above code fragment, which of the following holds the value of the current iteration number?

- a. tbl_roast c. COUNT
b. i d. lv_tot_num

____ 60. Which of the following dictates exactly how many times the loop should run in the opening LOOP clause?
a. CASE c. FOR loop
b. WHILE loop d. Basic loop

____ 61. Which of the following code fragments would not raise an error?

a. BEGIN
 FOR i IN 1..10 LOOP
 DBMS_OUTPUT.PUT_LINE(i);
 END LOOP;
END;

b. BEGIN
 FOR i IN 1..10 LOOP
 DBMS_OUTPUT.PUT_LINE(c)
 END LOOP
END;

```
c. BEGIN
    FOR i IN 1..10
        DBMS_OUTPUT.PUT_LINE(i);
    END LOOP;
END;
```

```
d. BEGIN
    FOR i IN 1..10 LOOP
        DBMS_OUTPUT.PUT_LINE
    END LOOP
END;
```

- ___ 62. Which of the following can be referenced in the loop but cannot be assigned a value because it is controlled by the loop?
- | | |
|-------------------------|---------------------------|
| a. The counter variable | c. The INSERT INTO clause |
| b. The IN clause | d. The FOR clause |

- ___ 63. Which of the following code fragments is correct?

```
a. FOR i IN 1..tbl.COUNT
    lv_tot_num := lv_tot_num + tbl_roast(i);
END LOOP;
```

```
b. FOR i IN 1..tbl.COUNT LOOP
    lv_tot_num := lv_tot_num + tbl_roast(i)
END LOOP
```

```
c. FOR i IN 1..tbl.COUNT LOOP
    lv_tot_num = lv_tot_num + tbl_roast(i);
END LOOP;
```

```
d. FOR i IN 1..tbl.COUNT LOOP
    lv_tot_num := lv_tot_num + tbl_roast(i);
END LOOP;
```

- ___ 64. Why would the following code raise an error?

```
IF rec_order.state = 'VA' THEN
    lv_tax_num := rec_order.sub * .06;
ELSEIF rec_order.state = 'ME' THEN
    lv_tax_num := rec_order.sub * .05;
ELSE
    lv_tax_num := rec_order.sub * .04;
END IF;
```

- | | |
|------------------------------------|----------------------------|
| a. Semicolon after THEN is omitted | c. ELSEIF is not a keyword |
| b. No space between ELSE and IF | d. “:=” should be “=” |

- ___ 65. IF rec_order.state = 'VA' THEN
 lv_tax_num := rec_order.sub * .06;
 ELSIF rec_order.state = 'ME' THEN
 lv_tax_num := rec_order.sub * .05;
 ELSE

```
lv_tax_num := rec_order.sub * .04;  
END IF;
```

Which of the clauses in the code fragment above would not cause the IF statement to raise an error if it were excluded?

- a. ELSE
- b. IF
- c. END IF
- d. THEN

Completion

Complete each statement.

- 66. The _____ section of the PL/SQL block contains code that creates variables, cursors, and types.
- 67. A(n) _____ variable can hold only a single value.
- 68. _____ statements are used to put or change values in variables.
- 69. The _____ section of a PL/SQL block contains handlers that allow you to control what the application will do if an error occurs during the executable statements in the BEGIN section.
- 70. To declare a variable, you must supply a variable name and _____.
- 71. The code *order NUMBER(2) := 6;* is an example of a(n) _____.
- 72. The _____ option can be added to the variable declaration to require the variable to always contain a particular value within the block.
- 73. The _____ statement is a mechanism that allows the checking of a condition to determine whether or not statements should be processed.
- 74. The simple IF statement performs an action only if the condition is _____.
- 75. A(n) _____ statement does not use a selector but individually evaluates conditions that are placed in the WHEN clauses.
- 76. _____ are used for situations in which we need to repeat a line or lines of code within our block.
- 77. If the EXIT WHEN clause is not included in the basic loop, the result is the programmer's nightmare of a(n) _____.
- 78. By indicating a numeric range, the _____ dictates exactly how many times the loop should run in the opening LOOP clause.
- 79. The _____ loop is constructed with conditions in the LOOP statement to determine when the looping action begins and ends.

Essay

80. What is the purpose of the BEGIN section of a PL/SQL block?
81. What is the purpose of control structures?

ch02
Answer Section

TRUE/FALSE

- | | | |
|------------|--------|---------|
| 1. ANS: T | PTS: 1 | REF: 36 |
| 2. ANS: T | PTS: 1 | REF: 36 |
| 3. ANS: F | PTS: 1 | REF: 36 |
| 4. ANS: T | PTS: 1 | REF: 37 |
| 5. ANS: T | PTS: 1 | REF: 41 |
| 6. ANS: F | PTS: 1 | REF: 47 |
| 7. ANS: T | PTS: 1 | REF: 47 |
| 8. ANS: T | PTS: 1 | REF: 49 |
| 9. ANS: F | PTS: 1 | REF: 50 |
| 10. ANS: F | PTS: 1 | REF: 50 |
| 11. ANS: T | PTS: 1 | REF: 53 |
| 12. ANS: F | PTS: 1 | REF: 58 |
| 13. ANS: T | PTS: 1 | REF: 59 |
| 14. ANS: T | PTS: 1 | REF: 59 |
| 15. ANS: F | PTS: 1 | REF: 60 |
| 16. ANS: F | PTS: 1 | REF: 60 |
| 17. ANS: T | PTS: 1 | REF: 60 |
| 18. ANS: F | PTS: 1 | REF: 61 |
| 19. ANS: T | PTS: 1 | REF: 63 |

MODIFIED TRUE/FALSE

- | | | | |
|---------------------|--------|---------|----------------|
| 20. ANS: F, DECLARE | | | |
| | PTS: 1 | REF: 36 | |
| 21. ANS: T | | | PTS: 1 REF: 36 |
| 22. ANS: F, BEGIN | | | |
| | PTS: 1 | REF: 37 | |
| 23. ANS: T | | | PTS: 1 REF: 37 |
| 24. ANS: T | | | PTS: 1 REF: 37 |
| 25. ANS: F, scalar | | | |
| | PTS: 1 | REF: 38 | |
| 26. ANS: T | | | PTS: 1 REF: 47 |
| 27. ANS: F, WHEN | | | |
| | PTS: 1 | REF: 55 | |
| 28. ANS: T | | | PTS: 1 REF: 56 |
| 29. ANS: T | | | PTS: 1 REF: 57 |
| 30. ANS: T | | | PTS: 1 REF: 57 |

31. ANS: F, EXIT WHEN
PTS: 1 REF: 59
32. ANS: F, FOR
PTS: 1 REF: 61
33. ANS: F, static
PTS: 1 REF: 63

MULTIPLE CHOICE

34. ANS: C PTS: 1 REF: 36
35. ANS: A PTS: 1 REF: 36
36. ANS: B PTS: 1 REF: 37
37. ANS: A PTS: 1 REF: 37
38. ANS: D PTS: 1 REF: 37
39. ANS: D PTS: 1 REF: 39
40. ANS: D PTS: 1 REF: 41
41. ANS: C PTS: 1 REF: 42
42. ANS: D PTS: 1 REF: 44
43. ANS: B PTS: 1 REF: 44
44. ANS: B PTS: 1 REF: 46
45. ANS: D PTS: 1 REF: 47-48
46. ANS: C PTS: 1 REF: 50
47. ANS: B PTS: 1 REF: 50
48. ANS: B PTS: 1 REF: 55
49. ANS: C PTS: 1 REF: 56
50. ANS: C PTS: 1 REF: 57
51. ANS: B PTS: 1 REF: 57
52. ANS: A PTS: 1 REF: 57
53. ANS: D PTS: 1 REF: 58
54. ANS: A PTS: 1 REF: 59
55. ANS: A PTS: 1 REF: 59
56. ANS: C PTS: 1 REF: 59
57. ANS: D PTS: 1 REF: 60
58. ANS: C PTS: 1 REF: 60
59. ANS: B PTS: 1 REF: 61
60. ANS: C PTS: 1 REF: 61
61. ANS: A PTS: 1 REF: 61
62. ANS: A PTS: 1 REF: 62
63. ANS: D PTS: 1 REF: 63
64. ANS: C PTS: 1 REF: 64|65
65. ANS: A PTS: 1 REF: 64-65

COMPLETION

66. ANS: DECLARE
PTS: 1 REF: 36
67. ANS: scalar
PTS: 1 REF: 36
68. ANS: Assignment
PTS: 1 REF: 37
69. ANS: EXCEPTION
PTS: 1 REF: 37
70. ANS: data type
PTS: 1 REF: 37
71. ANS: assignment statement
PTS: 1 REF: 39
72. ANS: CONSTANT
PTS: 1 REF: 42
73. ANS: IF
PTS: 1 REF: 47
74. ANS: TRUE
PTS: 1 REF: 47
75. ANS: Searched CASE
PTS: 1 REF: 55
76. ANS: Loops
PTS: 1 REF: 57
77. ANS: infinite loop
PTS: 1 REF: 59
78. ANS: FOR loop
PTS: 1 REF: 61
79. ANS:
WHILE
FOR
PTS: 1 REF: 63

ESSAY

80. ANS:

The BEGIN section is the heart of the PL/SQL block in that it contains all the processing action, or programming logic. SQL is used for database queries and data manipulation. Conditional logic, such as IF statements, is used to make decisions on what action to take. Loops are used to repeat code and assignment statements are used to put or change values in variables.

PTS: 1 REF: 37

81. ANS:

Control structures provide the capability to perform conditional logic to determine which statements should be run, how many times the statements should be run, and the overall sequence of events.

PTS: 1 REF: 46|47