

CHAPTER 2 - BASIC PL/SQL BLOCK STRUCTURES

TRUE/FALSE

1. The term anonymous blocks refers to blocks of code that are not stored for reuse and do not exist after being executed.

ANS: T PTS: 1 REF: 37

2. The only required sections of a PL/SQL block are DECLARE and END.

ANS: T PTS: 1 REF: 37

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

ANS: F PTS: 1 REF: 38

4. Assignment statements are used to put or change the values of variables.

ANS: T PTS: 1 REF: 41

5. The keyword DEFAULT can be used in place of the := symbol to assign initial values to the variables within the declaration statement.

ANS: T PTS: 1 REF: 41

6. The LOOP statement is a mechanism that allows the checking of a condition to determine if statements should or should not be processed.

ANS: F PTS: 1 REF: 53

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.

ANS: T PTS: 1 REF: 53

8. With respect to processing efficiency, the less code that has to be processed, the faster the program runs.

ANS: T PTS: 1 REF: 49

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
```

ANS: F PTS: 1 REF: 56

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
```

ANS: F PTS: 1 REF: 56

11. The CASE statement begins with the keyword CASE followed by a selector that is typically a variable name.

ANS: T PTS: 1 REF: 63

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;
```

ANS: F PTS: 1 REF: 69

13. An infinite loop causes a program to loop indefinitely, disrupting the ability of the code to continue with any processing beyond the loop.

ANS: T PTS: 1 REF: 69

14. The EXIT WHEN clause ensures that a basic loop runs at least once.

ANS: T PTS: 1 REF: 70

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;
```

ANS: F PTS: 1 REF: 72

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;
```

ANS: F PTS: 1 REF: 72

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;
```

ANS: T PTS: 1 REF: 72

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;
```

ANS: F PTS: 1 REF: 73

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.

ANS: T PTS: 1 REF: 72

MODIFIED TRUE/FALSE

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

ANS: F, DECLARE

PTS: 1 REF: 38

2. A(n) scalar variable can hold only a single value. _____

ANS: T PTS: 1 REF: 39

3. The DECLARE section of the PL/SQL block contains all the processing action, or programming logic.

ANS: F, BEGIN

PTS: 1 REF: 38

4. The EXCEPTION section of a PL/SQL block contains handlers that allow you to control what the application will do if an error occurs. _____

ANS: T PTS: 1 REF: 38

5. Variables are named memory areas that hold values to allow retrieval and manipulation of values within your programs. _____

ANS: T PTS: 1 REF: 39

6. The common data types used for cursor variables include character, numeric, date, and Boolean.

ANS: F, scalar

PTS: 1 REF: 39

7. 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. _____

ANS: T PTS: 1 REF: 53

8. The Searched CASE statement does not use a selector but individually evaluates conditions that are placed in the WHERE clauses. _____

ANS: F, WHEN

PTS: 1 REF: 66

9. A(n) CASE expression evaluates conditions and returns a value in an assignment statement.

ANS: T PTS: 1 REF: 67

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

ANS: T PTS: 1 REF: 69

11. The basic loop uses the LOOP and END LOOP markers to begin and end the loop code.

ANS: T PTS: 1 REF: 69

12. If the WHERE clause is not included the basic loop, the result is the programmer's nightmare of the infinite loop. _____

ANS: F, EXIT WHEN

PTS: 1 REF: 69

13. The basic loop dictates exactly how many times the loop should run in the opening LOOP clause.

ANS: F, FOR

PTS: 1 REF: 69

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

ANS: F, static

PTS: 1 REF: 69

MULTIPLE CHOICE

1. The only required sections of a PL/SQL block are the ____ sections.
- a. BEGIN & DECLARE
 - b. DECLARE & EXCEPTION
 - c. BEGIN & END
 - d. EXCEPTION & END

ANS: C PTS: 1 REF: 38

2. The ____ section of a PL/SQL block contains code that creates variables, cursors, and types.
- a. DECLARE
 - b. BEGIN
 - c. EXCEPTION
 - d. END

ANS: A PTS: 1 REF: 38

3. ____ are used to change the values of variables.
- a. Loops
 - b. Assignment statements
 - c. Exceptions
 - d. Blocks

ANS: B PTS: 1 REF: 44

4. 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

ANS: A PTS: 1 REF: 38

5. ____ 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

ANS: D PTS: 1 REF: 39

6. 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;

ANS: D PTS: 1 REF: 41

7. 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;

ANS: D PTS: 1 REF: 41

8. 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;

ANS: C PTS: 1 REF: 48

9. 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

ANS: D PTS: 1 REF: 50

10. 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;
```

ANS: B PTS: 1 REF: 50

11. 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

ANS: B PTS: 1 REF: 53

12. 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;

ANS: D PTS: 1 REF: 53

13. 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;

ANS: C PTS: 1 REF: 56

14. 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

ANS: B PTS: 1 REF: 54

15. Which of the following does not use a selector, but individually evaluates conditions that are placed in WHEN clauses?

- | | |
|-----------------------|--------------------|
| a. Control statements | c. Loops |
| b. Searched CASE | d. CASE expression |

ANS: B PTS: 1 REF: 66

16. Which of the following evaluates conditions and returns a value in an assignment statement?

- | | |
|------------------|----------------------|
| a. Searched CASE | c. CASE expression |
| b. Basic loop | d. Control statement |

ANS: C PTS: 1 REF: 67

17. 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".

ANS: C PTS: 1 REF: 67

18. 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

ANS: B PTS: 1 REF: 69

19. 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

ANS: A PTS: 1 REF: 69

20. 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;

ANS: D PTS: 1 REF: 69

21. If the EXIT WHEN clause is not included in a basic loop, then the result is a(n) ____.
- a. infinite loop
 - b. exception
 - c. RAISE_APPLICATION_ERROR
 - d. SQLCODE error

ANS: A PTS: 1 REF: 69

22. Which of the following clauses ensures that a basic loop runs at least once?
- a. EXIT WHEN
 - b. WHERE
 - c. CASE
 - d. LOOP

ANS: A PTS: 1 REF: 69

23. 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);

ANS: C PTS: 1 REF: 69

24. 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;

ANS: D PTS: 1 REF: 72

25. 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
- b. 4
- c. 5
- d. 6

ANS: C PTS: 1 REF: 72

26. 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
- b. i
- c. COUNT
- d. lv_tot_num

ANS: B PTS: 1 REF: 73

27. Which of the following dictates exactly how many times the loop should run in the opening LOOP clause?

- a. CASE
- b. WHILE loop
- c. FOR loop
- d. Basic loop

ANS: C PTS: 1 REF: 73

28. 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;

ANS: A PTS: 1 REF: 73

29. 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
 - b. The IN clause
 - c. The INSERT INTO clause
 - d. The FOR clause

ANS: A PTS: 1 REF: 73

30. 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;

ANS: D PTS: 1 REF: 73

31. 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
- b. No space between ELSE and IF
- c. ELSEIF is not a keyword
- d. “:=” should be “=”

ANS: C PTS: 1 REF: 56

32. 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

ANS: A

PTS: 1

REF: 54

COMPLETION

1. The _____ section of the PL/SQL block contains code that creates variables, cursors, and types.

ANS: DECLARE

PTS: 1

REF: 38

2. A(n) _____ variable can hold only a single value.

ANS: scalar

PTS: 1

REF: 39

3. _____ statements are used to put or change values in variables.

ANS: Assignment

PTS: 1

REF: 41

4. 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.

ANS: EXCEPTION

PTS: 1

REF: 38

5. To declare a variable, you must supply a variable name and _____.

ANS: data type

PTS: 1

REF: 41

6. The code *order* *NUMBER(2) := 6;* is an example of a(n) _____.

ANS: assignment statement

PTS: 1

REF: 41

7. The _____ option can be added to the variable declaration to require the variable to always contain a particular value within the block.

ANS: CONSTANT

PTS: 1

REF: 48

8. The _____ statement is a mechanism that allows the checking of a condition to determine whether or not statements should be processed.

ANS: IF

PTS: 1 REF: 53

9. The simple IF statement performs an action only if the condition is _____.

ANS: TRUE

PTS: 1 REF: 53

10. A(n) _____ statement does not use a selector but individually evaluates conditions that are placed in the WHEN clauses.

ANS: Searched CASE

PTS: 1 REF: 66

11. _____ are used for situations in which we need to repeat a line or lines of code within our block.

ANS: Loops

PTS: 1 REF: 69

12. If the EXIT WHEN clause is not included in the basic loop, the result is the programmer's nightmare of a(n) _____.

ANS: infinite loop

PTS: 1 REF: 69

13. By indicating a numeric range, the _____ dictates exactly how many times the loop should run in the opening LOOP clause.

ANS: FOR loop

PTS: 1 REF: 73

14. The _____ loop is constructed with conditions in the LOOP statement to determine when the looping action begins and ends.

ANS:
WHILE
FOR

PTS: 1 REF: 72-73

ESSAY

1. What is the purpose of the BEGIN section of a PL/SQL block?

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: 38

2. What is the purpose of control structures?

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: 53