PlSQL feedback midterm semister 1 part2

Test: Mid Term Exam Semester 1 - Part II

Review your answers, feedback, and question scores below. An asterisk (*) indicates a correct answer.

The Mid Term Exam for Semester 1 is presented to you as two exams. This is Part II of the Mid Term Exam for Semester 1.

Section 4

1. Examine the following code:

```
DECLARE
    a BOOLEAN := TRUE;
    b BOOLEAN := FALSE;
    c BOOLEAN := TRUE;
    d BOOLEAN := FALSE;
    game char(4) := 'lost';
BEGIN
    IF ((a AND b) AND (c OR d))
    THEN game := 'won';
    END IF;
```

What is the value of GAME at the end of this block?

Mark for Review

(1) Points

NULL

won'

lost' (*)

False

Incorrect. Refer to Section 4.

2. What is the correct form of a compound IF statement? Mark for Review

(1) Points

```
    IF condition
THEN statement1
ELSE statement 2;
```

```
    IF condition
THEN statement1
ELSE statement 2;
END IF;
```

```
    IF condition;
THEN statement1;
ELSE statement2;
END IF;
```

```
    IF condition THEN statement1;
ELSE statement2;
END IF;
```

(*)

Incorrect. Refer to Section 4.

3. How many ELSIF statements are you allowed to have in a compound IF statement?
Mark for Review
(1) Points

Only one

As many as you want (*)

They must match the same number as the number of ELSE statements.

None; the command is ELSE IF;

Incorrect. Refer to Section 4.

4. Examine the following code:

```
DECLARE
    a VARCHAR2(6) := NULL;
    b VARCHAR2(6) := NULL;
BEGIN
    IF a = b THEN
       DBMS_OUTPUT.PUT_LINE('EQUAL');
    ELSIF a != b THEN
       DBMS_OUTPUT.PUT_LINE('UNEQUAL');
    ELSE
       DBMS_OUTPUT.PUT_LINE('OTHER');
    END IF;
END;
```

Which word will be displayed?
Mark for Review
(1) Points

UNEQUAL

EQUAL

Nothing will be displayed

OTHER (*)

Incorrect. Refer to Section 4.

5. You need to execute a set of statements 10 times, increasing a counter by 1 each time. Which of the following PL/SQL constructs can do this? (Choose three)
Mark for Review
(1) Points

(Choose all correct answers)

IF ... THEN ... ELSE

A WHILE loop (*)

CASE ... WHEN ... THEN

A FOR loop (*)

A basic loop (*)

Incorrect. Refer to Section 4.

6. In the following code fragment, you want to exit from the outer loop at Line A if v_number = 6. Which statement would you write on Line A?

```
<<big_loop>>
WHILE condition_1 LOOP
    <<small_loop>>
    FOR i IN 1..10 LOOP
       DBMS_OUTPUT.PUT_LINE(i);
       -- Line A
    END LOOP;
END LOOP;
```

Mark for Review
(1) Points

IF v_number = 6 THEN EXIT;

EXIT outer_loop WHEN v_number = 6;

EXIT big_loop WHEN v_number = 6; (*)

EXIT small_loop WHEN v_number = 6;

Incorrect. Refer to Section 4.

7. You want to display multiplication tables for numbers up to 12. The display should look like this:

```
1 x 1 = 1
1 x 2 = 2
.....
1 x 12 = 12
2 x 1 = 2
2 x 2 = 4
.....
2 x 12 = 24
3 x 1 = 3
.....
.....
12 x 12 = 144
```

Which of the following is an efficient way to do this in PL/SQL? Mark for Review
(1) Points

Use two nested FOR loops. (*)

Store all the numbers from 1 to 144 in a table, then fetch and display them using a cursor.

Create a function which accepts two numbers as IN parameters and returns their product. Invoke the function 144 times.

Write an anonymous block which contains 144 calls to DBMS_OUTPUT, each looking

like: DBMS_OUTPUT.PUT_LINE('7 x 9 = 63');

Incorrect. Refer to Section 4.

8. Examine the following code:

```
DECLARE
    v_outer_count NUMBER := 1;
    v_inner_count NUMBER := 1;
BEGIN
    LOOP
       LOOP
          v_inner_count := v_inner_count + 1;
          EXIT WHEN v_inner_count > 5; -- Line A
       END LOOP;
       v_outer_count := v_outer_count + 1;
       EXIT WHEN v_outer_count > 3;
    END LOOP;
END;
```

What happens at Line A when the value of V_INNER_COUNT equals 6?
Mark for Review
(1) Points

Both loops are exited and the block's execution is terminated.

The inner loop is exited but the outer loop continues execution. (*)

The outer loop is exited but the inner loop continues execution.

An error condition is returned.

Incorrect. Refer to Section 4.

9. Examine the following code:

```
BEGIN
FOR i IN 1..5 LOOP
FOR j IN 1..8 LOOP
EXIT WHEN j = 7;
DBMS_OUTPUT.PUT_LINE(i || j);
END LOOP;
END LOOP;
END;
```

How many lines of output will be displayed when this code is executed? Mark for Review
(1) Points

35

6

30 (*)

40

Incorrect. Refer to Section 4.

10. The EXIT statement can be located anywhere inside a basic loop. True or

False? Mark for Review
(1) Points

True (*)

False

Correct

Test: Mid Term Exam Semester 1 - Part II

Review your answers, feedback, and question scores below. An asterisk (*) indicates a correct answer.

The Mid Term Exam for Semester 1 is presented to you as two exams. This is Part II of the Mid Term Exam for Semester 1.

Section 4

11. Which one of these is NOT a kind of loop? Mark for Review
(1) Points

ASCENDING loop (*)

FOR loop

Basic loop

WHILE loop

Correct

12. Examine the following block:

```
DECLARE
    v_counter PLS_INTEGER := 1;
BEGIN
    LOOP
       DBMS_OUTPUT.PUT_LINE(v_counter);
       v_counter := v_counter + 1;
    EXIT WHEN v_counter = 5;
    END LOOP;
END;
```

What is the last value of V_COUNTER that is displayed?
Mark for Review
(1) Points

5

6

4 (*)

This is an infinite loop; the loop will never finish.

Incorrect. Refer to Section 4.

13. A PL/SQL block contains the following code:

```
v_counter := 1;
LOOP
    EXIT WHEN v_counter=5;
END LOOP;
v_counter := v_counter + 1;
```

What is the value of V_COUNTER after the loop is finished?
Mark for Review
(1) Points

5

6

1

This is an infinite loop; the loop will never finish. (*)

Incorrect. Refer to Section 4.

14. A PL/SQL block contains the following code:

```
v_counter := 1;
LOOP
EXIT WHEN v_counter = 5;
    v_counter := v_counter + 1;
END LOOP;
```

What is the value of V_COUNTER after the loop is finished?
Mark for Review
(1) Points

5 (*)

6

1

This is an infinite loop; the loop will never finish.

Correct

15. In a WHILE loop, the statements inside the loop must execute at least once. True or False? Mark for Review
(1) Points

True

False (*)

Incorrect. Refer to Section 4.

16. Which of the following blocks produces the same output as this block?

```
BEGIN
    FOR i in 1 .. 3 LOOP
       DBMS_OUTPUT.PUT_LINE(i);
    END LOOP;
END;
```

Mark for Review
(1) Points

```
    DECLARE
    i PLS_INTEGER := 0;
BEGIN
    WHILE i<3 LOOP
       DBMS_OUTPUT.PUT_LINE(i);
       i := i + 1;
    END LOOP;
END;
```

```
    DECLARE
    i PLS_INTEGER := 0;
BEGIN
    WHILE i<3 LOOP
       i := i + 1;
       DBMS_OUTPUT.PUT_LINE(i);
    END LOOP;
END;
```

(*)

```
    DECLARE
    i PLS_INTEGER := 0;
BEGIN
    WHILE i<3 LOOP
       DBMS_OUTPUT.PUT_LINE(i);
    END LOOP;
    i := i+ 1;
END;
```

Incorrect. Refer to Section 4.

17. Which statement best describes when a WHILE loop shouild be used? Mark for Review
(1) Points

When the number of iterations is known

When repeating a sequence of statements until the controlling condition is no longer true (*)

When assigning a value to a Boolean variable

When testing whether a variable is null

Incorrect. Refer to Section 4.

18. What will happen when the following code is executed?

```
BEGIN
FOR i in 1 ..3 LOOP
    DBMS_OUTPUT.PUT_LINE (i);
    i := i + 1;
END LOOP;
END;
```

Mark for Review
(1) Points

It will display 1, 2, 3.

It will display 2, 3, 4.

It will result in an error because you cannot modify the counter in a FOR loop. (*)

It will result in an error because the counter was not explicitly declared.

Incorrect. Refer to Section 4.

19. Which statement best describes when a FOR loop should be used? Mark for Review
(1) Points

When the number of iterations is known (*)

When testing the value in a Boolean variable

When the controlling condition must be evaluated at the start of each iteration

Incorrect. Refer to Section 4.

20. Examine the following code:

```
DECLARE
v_bool BOOLEAN := FALSE;
v_counter NUMBER(4) := 0;
BEGIN
... Line A
?
END;
```

Which of the following is NOT valid at line A?
Mark for Review
(1) Points

WHILE NOT v_boolean LOOP

WHILE v_boolean AND v_counter < 6 LOOP

WHILE v_counter > 8 LOOP

WHILE v_counter IN 1..5 LOOP (*)

Incorrect. Refer to Section 4.

Test: Mid Term Exam Semester 1 - Part II

Review your answers, feedback, and question scores below. An asterisk (*) indicates a correct answer.

The Mid Term Exam for Semester 1 is presented to you as two exams. This is Part II of the Mid Term Exam for Semester 1.

Section 4

21. You want to display a message which depends on the value of v_grade: if v_grade = 'A' display 'Very Good', if v_grade = 'B' then display 'Good', and so on.

```
DECLARE
    v_grade CHAR(1);
BEGIN
    CASE v_grade
```

The next line should be
Mark for Review
(1) Points

```
    WHEN 'A' THEN (*)

    WHEN v_grade = 'A' THEN

    WHEN 'A' THEN;

    IF 'A' THEN
```

Correct

22. You want to assign a value to v_result which depends on the value of v_grade: if v_grade = 'A' set v_result to 'Very Good' and so on.

```
DECLARE
    v_grade CHAR(1);
    v_result VARCHAR2(10);
BEGIN
    v_result :=
       CASE v_grade
```

The next line should be
Mark for Review
(1) Points

```
    WHEN v_grade = 'A' THEN 'Very Good'

    WHEN 'A' THEN 'Very Good';
```

WHEN 'A' THEN v_result := 'Very Good';

WHEN 'A' THEN 'Very Good' (*)

Incorrect. Refer to Section 4.

23. What will be the value of v_result after the following code is executed?

```
DECLARE
    v_grade CHAR(1) := NULL;
    v_result VARCHAR2(10);
BEGIN
    CASE v_grade
       WHEN 'A' THEN v_result := 'Very Good';
       WHEN 'F' THEN v_result := 'Poor';
       ELSE v_result := 'In Between';
    END;
END;
```

Mark for Review
(1) Points

Poor

In Between (*)

Null

Very Good

Incorrect. Refer to Section 4.

24. What will be the value of variable c after the following code is executed?

```
DECLARE
    a BOOLEAN := TRUE;
    b BOOLEAN := FALSE;
    c NUMBER;
BEGIN
    c :=
        CASE
            WHEN a AND b THEN 10
            WHEN NOT a THEN 20
            WHEN a OR b THEN 30
            ELSE 40
        END;
END;
```

Mark for Review
(1) Points

30 (*)

20

40

10

Correct

25. What will be the value of v_sal_desc after the following code is executed?

```
DECLARE
    v_salary NUMBER(6,2) := NULL;
    v_sal_desc VARCHAR2(10);
BEGIN
    CASE
       WHEN v_salary < 10000 THEN v_sal_desc := 'Low Paid';
       WHEN v_salary >= 10000 THEN v_sal_desc := 'High Paid';
    END CASE;
END;
```

Mark for Review
(1) Points

High Paid

Low Paid

Null

The code will fail and return an exception (*)

Incorrect. Refer to Section 4.

Section 5

26. An implicit cursor can be used for a multiple-row SELECT statement. True or False? Mark for Review
(1) Points

True

False (*)

Incorrect. Refer to Section 5.

27. For which type of SQL statement must you use an explicit cursor? Mark for Review
(1) Points

DML statements that process more than one row.

Queries that return more than one row. (*)

Data Definition Language (DDL) statements.

Queries that return a single row.

Correct

28. After a cursor has been closed, it can be opened again in the same PL/SQL block. True or False? Mark for Review

PlSQL feedback midterm semister 1 part2

(1) Points

True (*)

False

Correct

29. Which of these constructs can be used to fetch multiple rows from a cursor's active set? Mark for Review

(1) Points

A CASE statement

An IF .... ELSE statement

A basic loop which includes FETCH and EXIT WHEN statements (*)

A basic loop which includes OPEN, FETCH and CLOSE statements

Incorrect. Refer to Section 5.

30. What will happen when the following code is executed?

```
DECLARE CURSOR emp_curs IS
    SELECT salary FROM employees;
    v_salary employees.salary%TYPE;
BEGIN
    OPEN emp_curs;
    FETCH emp_curs INTO v_salary;
    CLOSE emp_curs;
    FETCH emp_curs INTO v_salary;
END;
```

Mark for Review

(1) Points

The block will fail and an INVALID_CURSOR exception will be raised. (*)

The first employee row will be fetched twice.

The first two employee rows will be fetched.

The block will fail and a TOO_MANY_ROWS exception will be raised.

Incorrect. Refer to Section 5.

PlSQL feedback midterm semister 1 part2

Review your answers, feedback, and question scores below. An asterisk (*) indicates a correct answer.

The Mid Term Exam for Semester 1 is presented to you as two exams. This is Part II of the Mid Term Exam for Semester 1.

Section 5

31. The employees table contains 20 rows. What will happen when the following code is executed?

```
DECLARE
   CURSOR emp_curs IS
      SELECT job_id FROM employees;
    v_job_id employees.job_id%TYPE;
BEGIN
    OPEN emp_curs;
    LOOP
       FETCH emp_curs INTO v_job_id;
       DBMS_OUTPUT.PUT_LINE(v_job_id);
       EXIT WHEN emp_curs%NOTFOUND;
    END LOOP;
    CLOSE emp_curs;
END;
```

Mark for Review
(1) Points

20 job_ids will be displayed.

The block will fail and an error message will be displayed.

21 rows of output will be displayed; the first job_id will be displayed twice.

21 rows of output will be displayed; the last job_id will be displayed twice. (*)

Incorrect. Refer to Section 5.

32. Place the following statements in the correct sequence:

```
OPEN my_curs;
CLOSE my_curs;
CURSOR my_curs IS SELECT my_column FROM my_table;
FETCH my_curs INTO my_variable;
```

Mark for Review
(1) Points

C,D,A,B

C,A,D,B (*)

A,C,D,B

C,A,B,D

Incorrect. Refer to Section 5.

33. Which of these statements about implicit cursors is NOT true? Mark for Review
(1) Points

They are declared automatically by Oracle for single-row SELECT statements.

They are declared automatically by Oracle for all DML statements.

They are declared by the PL/SQL programmer. (*)

They are opened and closed automatically by Oracle.

Incorrect. Refer to Section 5.

34. You want to display each row from the DEPARTMENTS table, and immediately underneath it, a list of all EMPLOYEES in that department. Which of the following is a good way to do this? Mark for Review
(1) Points

Use a single cursor, declared as SELECT * FROM employees GROUP BY department_id;

Use two cursors, one for each of the two tables. Declare the EMPLOYEES cursor with a parameter for the DEPARTMENT_ID. (*)

Write a SELECT statement which JOINs the two tables, and use CONNECT BY PRIOR and LEVEL to display the rows in the correct order.

Use a single cursor with a cursor FOR loop.

Change the physical model so that all employee and department data is in a single table.

Incorrect. Refer to Section 5.

35. Which of the following is a good reason to declare and use multiple cursors in a single PL/SQL block? Mark for Review
(1) Points

Multiple cursors improve performance. They are faster than using a single cursor.

Multiple cursors use less memory than a single cursor.

Multiple cursors allow us to fetch rows from two or more related tables without using a JOIN. (*)

Multiple cursors are the only way to use cursors with parameters.

Multiple cursors can be opened many times, while a single cursor can be opened only once.

Incorrect. Refer to Section 5.

36. What is wrong with the following code?

```
DECLARE
CURSOR emp_curs(p_dept_id NUMBER) IS
```

```
SELECT * FROM employees WHERE department_id = p_dept_id;
BEGIN
FOR dept_rec IN (SELECT * FROM departments) LOOP
DBMS_OUTPUT.PUT_LINE(dept_rec.department_name);
FOR emp_rec IN emp_curs(dept_rec.department_id) LOOP
DBMS_OUTPUT.PUT_LINE(emp_rec.last_name);
END LOOP;
END LOOP;
END;
```

Mark for Review
(1) Points

The DEPARTMENTS cursor must be declared with a parameter.

You cannot use a cursor with a subquery in nested loops.

You cannot use two different kinds of loop in a single PL/SQL block.

EMP_CURS should not be DECLAREd explicitly; it should be coded as a subquery in a cursor FOR loop.

Nothing is wrong. The block will execute successfully and display all departments and the employees in those departments. (*)

Incorrect. Refer to Section 5.

37. What is wrong with the following code?

```
DECLARE
    CURSOR dept_curs IS SELECT * FROM departments;
BEGIN
    FOR dept_rec IN dept_curs LOOP
       DBMS_OUTPUT.PUT_LINE(dept_curs%ROWCOUNT || dept_rec.department_name);
    END LOOP;
    DBMS_OUTPUT.PUT_LINE(dept_rec.department_id);
END;
```

Mark for Review
(1) Points

The cursor DEPT_CURS has not been opened.

The implicitly declared record DEPT_REC cannot be referenced outside the cursor FOR loop. (*)

You cannot use %ROWCOUNT with a cursor FOR loop.

The cursor DEPT_CURS has not been closed.

Nothing is wrong, this code will execute successfully.

Incorrect. Refer to Section 5

38. There are no employees in department_id 75.
Which of the following is NOT a valid cursor FOR loop with a subquery? Mark for Review
(1) Points

```
    FOR emp_rec IN
   (SELECT last_name, salary FROM employees) LOOP ...
```

```
FOR emp_rec IN
(SELECT * FROM employees) LOOP ...


FOR emp_rec IN
(SELECT last_name, salary FROM employees ORDER BY last_name) LOOP ...


FOR emp_rec IN
(SELECT * FROM employees WHERE department_id = 75) LOOP ...
```

None of the above. They are all valid. (*)

Incorrect. Refer to Section 5

39. When using a cursor FOR loop, OPEN, CLOSE and FETCH statements should not be explicitly coded. True or False? Mark for Review
(1) Points

True (*)

False

Incorrect. Refer to Section 5

40. Examine the following code:

```
DECLARE
CURSOR c IS SELECT * FROM employees FOR UPDATE;
c_rec c%ROWTYPE;
BEGIN
OPEN c;
FOR i IN 1..20 LOOP
FETCH c INTO c_rec;
IF i = 6 THEN
UPDATE employees SET first_name = 'Joe'
WHERE CURRENT OF c;
END IF;
END LOOP;
CLOSE c;
END;
```

Which employee row or rows will be updated when this block is executed? Mark for Review
(1) Points

The first 6 fetched rows will be updated.

No rows will be updated because you locked the rows when the cursor was opened.

The 6th fetched row will be updated. (*)

The block will not compile because the cursor should have been declared .... FOR UPDATE WAIT 5;

None of the above.

Incorrect. Refer to Section 5.

Test: Mid Term Exam Semester 1 - Part II

Review your answers, feedback, and question scores below. An asterisk (*) indicates a correct answer.

The Mid Term Exam for Semester 1 is presented to you as two exams. This is Part II of the Mid Term Exam for Semester 1.

Section 5

41. User TOM has locked a row in the WORKERS table. Now, user DICK wants to open the following cursor:

```
CURSOR c IS
SELECT * FROM workers FOR UPDATE NOWAIT;
```

What will happen when DICK opens the cursor and tries to fetch rows? Mark for Review
(1) Points

TOM's session is rolled back. DICK's session successfully fetches rows from the cursor.

DICK's session waits indefinitely.

Both sessions wait for a few seconds; then the system breaks all locks and both sessions raise an exception.

DICK's session immediately raises an exception. (*)

The c%NOWAIT attribute is set to TRUE.

Incorrect. Refer to Section 5.

42. Consider the following cursor:

```
CURSOR c IS
    SELECT e.last_name, e.salary, d.department_name
    FROM employees e JOIN departments d
    USING(department_id)
    WHERE e.last_name='Smith'
    FOR UPDATE;
```

When the cursor is opened and rows are fetched, what is locked?
Mark for Review
(1) Points

The whole EMPLOYEES table is locked.

In the EMPLOYEES table, only the 'Smith' rows are locked. Nothing in the

DEPARTMENTS table is locked.

Each 'Smith' row is locked and Smith's matching rows in DEPARTMENTS are locked. No other rows are locked in either table. (*)

The whole EMPLOYEES and DEPARTMENTS tables are locked.

Nothing is locked because the cursor was not declared with NOWAIT.

Incorrect. Refer to Section 5.

43. What is one of the advantages of using parameters with a cursor? Mark for Review
(1) Points

You can use a cursor FOR loop.

You can declare the cursor FOR UPDATE.

You do not need to DECLARE the cursor at all.

You can use a single cursor to fetch a different set of rows each time the cursor is opened. (*)

It will execute much faster than a cursor without parameters.

Incorrect. Refer to Section 5.

44. Examine the following declaration of a cursor with a parameter. What should be coded at Point A? DECLARE
CURSOR emp_curs(-- Point A --) IS
SELECT * FROM employees
WHERE job_id = p_job_id;
Mark for Review
(1) Points

p_job_id

ST_CLERK'

p_job_id VARCHAR2(25)

p_job_id VARCHAR2 (*)

job_id VARCHAR2

Incorrect. Refer to Section 5.

45. Examine the following code fragment:

```
DECLARE
    CURSOR emp_curs IS
       SELECT first_name, last_name FROM employees;
    v_emp_rec emp_curs%ROWTYPE;
BEGIN
    ...
    FETCH emp_curs INTO v_emp_rec;
    DBMS_OUTPUT.PUT_LINE(... Point A ...);
```

```
   ...
```

To display the fetched last name, what should you code at Point A?
Mark for Review
(1) Points

v_emp_rec.last_name (*)

v_emp_rec(last_name)

v_emp_rec

last_name

None of the above

Incorrect. Refer to Section 5.

46. The following cursor has been declared:

```
CURSOR emp_curs IS
   SELECT first_name, last_name, job_id, salary
   FROM employees;
```

Which of the following correctly declares a composite record with the same structure as the cursor?
Mark for Review
(1) Points

emp_rec emp_rec%ROWTYPE;

emp_rec emp_curs%TYPE;

emp_rec emp_curs%ROWTYPE; (*)

emp_rec cursor%ROWTYPE;

Incorrect. Refer to Section 5.

47. The DEPARTMENTS table contains four columns. Examine the following code:

```
DECLARE
    CURSOR dept_curs IS
       SELECT * FROM departments;
    v_dept_rec dept_curs%ROWTYPE;
BEGIN
    OPEN dept_curs;
    FETCH dept_curs INTO v_dept_rec;
    ...
```

Which one of the following statements is true?
Mark for Review
(1) Points

v_dept_rec contains the first four rows of the departments table.

The FETCH will fail because the structure of v_dept_rec does not match the structure of the cursor.

v_dept_rec contains the first row of the departments table. (*)

The block will fail because the declaration of v_dept_rec is invalid.

Incorrect. Refer to Section 5.

48. Assume that you have declared a cursor called C_EMP. Which of the following statements about C_EMP is correct? (Choose two.) Mark for Review
(1) Points

(Choose all correct answers)

You can use c_emp%NOTFOUND to exit a loop. (*)

You can fetch rows when c_emp%ISOPEN evaluates to FALSE.

You can use c_emp%ROWCOUNT to return the number of rows returned by the cursor so far. (*)

You can use c_emp%FOUND after the cursor is closed.

Incorrect. Refer to Section 5.

49. Which of the following cursor attributes evaluates to TRUE if the cursor is open? Mark for Review
(1) Points

%ISOPEN (*)

%NOTFOUND

%FOUND

%ROWCOUNT

Correct

50. Which of the following statements about the %ISOPEN cursor attribute is true? Mark for Review
(1) Points

You can issue the %ISOPEN cursor attribute only when a cursor is open.

You can issue the %ISOPEN cursor attribute only when more than one record is returned.

You can issue the %ISOPEN cursor attribute when a cursor is open or closed. (*)

If a cursor is open, then the value of %ISOPEN is false.

Incorrect. Refer to Section 5.

PlSQL feedback midterm semister 1 part2

Review your answers, feedback, and question scores below. An asterisk (*) indicates a correct answer.

Section 9

1. Which of the following SQL statements can be included in a PL/SQL block only by using Dynamic SQL? (Choose two.) Mark for Review
(1) Points

(Choose all correct answers)

DELETE

SAVEPOINT

ALTER (*)

SELECT ..... FOR UPDATE NOWAIT

GRANT (*)

Incorrect. Refer to Section 9.

2. The easiest way to include DDL statements in a PL/SQL block is to use the DBMS_SQL package. True or False? Mark for Review
(1) Points

True

False (*)

Incorrect. Refer to Section 9.

3. A SQL statement can pass through several stages. Which of the following is NOT one of these stages? Mark for Review
(1) Points

BIND

FETCH

PARSE

RETURN (*)

EXECUTE

Incorrect. Refer to Section 9.

4. A public packaged procedure contains the following SQL statement:
UPDATE employees SET salary = salary * 1.1;
When is this SQL statement parsed? Mark for Review
(1) Points

When the package specification is created

When the package body is created (*)

When the package header is loaded into memory.

When the package is loaded into memory.

Only the first time the procedure is executed.

Incorrect. Refer to Section 9.

5. Package HRPACK contains the following public function:

```
FUNCTION empfunc (p_deptno NUMBER) RETURN NUMBER IS
BEGIN
    UPDATE employees SET salary = salary * 1.1
       WHERE department_id = p_deptno;
    RETURN SQL%ROWCOUNT;
END empfunc;
```

What will happen when the following SQL statement is executed?

```
SELECT department_name, hrpack.empfunc(department_id)
FROM departments;
```

Mark for Review
(1) Points

The SELECT will fail because you cannot return SQL%ROWCOUNT from a packaged function.

The SELECT will fail because you cannot call packaged functions from within a SQL statement.

The SELECT will fail because you cannot execute a DML statement from within a query.

The SELECT will succeed because it is referencing a different table from the function. (*)

Incorrect. Refer to Section 9.

6. Examine the following code:

```
CREATE OR REPLACE PACKAGE emppack IS
    PROCEDURE upd_emp (p_empno IN NUMBER, p_salary IN NUMBER);
END emppack;
CREATE OR REPLACE PACKAGE BODY emppack IS
    -- Line A
    PROCEDURE upd_emp (p_empno IN NUMBER, p_salary IN NUMBER) IS
       BEGIN
          IF NOT sal_ok(p_salary) THEN
             RAISE_APPLICATION_ERROR(-20201,'Invalid salary');
          END IF;
    END upd_emp;
    FUNCTION sal_ok(pf_salary NUMBER) RETURN BOOLEAN IS
       BEGIN
          IF pf_salary > 50000 THEN RETURN FALSE;
          ELSE RETURN TRUE;
          END IF;
    END sal_ok;
```

```
END emppack;
```

What must be coded at Line A for this package to compile successfully?
Mark for Review
(1) Points

FUNCTION sal_ok;

FUNCTION sal_ok(pf_salary NUMBER);

FUNCTION sal_ok(pf_salary NUMBER) RETURN BOOLEAN; (*)

PROCEDURE upd_emp (p_empno IN NUMBER, p_salary IN NUMBER);

Nothing is needed at Line A

Correct

7. Examine the following package code:

```
CREATE OR REPLACE PACKAGE ol_pack IS
    PROCEDURE subprog (p1 IN VARCHAR2, p2 IN NUMBER);
    PROCEDURE subprog (param1 IN CHAR, param2 IN NUMBER);
    FUNCTION subprog (param1 IN VARCHAR2, param2 IN NUMBER) RETURN DATE;
END ol_pack;
```

Which of the following calls will be successful? (Choose two.)
Mark for Review
(1) Points

(Choose all correct answers)

ol_pack.subprog('Jane',30);

ol_pack.subprog(param1=>'Jane',param2=>30); (*)

v_number := ol_pack.subprog(p1=>'Jane');

v_date := ol_pack.subprog('Jane',30); (*)

Incorrect. Refer to Section 9.

8. Which two of these declarations cannot be in the same package specification?

```
PROCEDURE myproc (p1 NUMBER, p2 VARCHAR2);
PROCEDURE myproc (p1 VARCHAR2, p2 NUMBER);
PROCEDURE myproc (p1 NUMBER, p2 CHAR);
PROCEDURE myproc (p1 NUMBER);
```

Mark for Review
(1) Points

1 and 2

1 and 3 (*)

2 and 3

3 and 4

1 and 4

Correct

9. Package MULTIPACK declares the following global variable:
g_myvar NUMBER;
User DICK executes the following:
multipack.g_myvar := 45;

User HAZEL now connects to the database. Both users immediately execute:

```
BEGIN
    DBMS_OUTPUT.PUT_LINE(multipack.g_myvar);
END;
```

What values will Dick and Hazel see?
Mark for Review
(1) Points

Dick: 45, Hazel: 45

Dick: 45, Hazel: 0

Dick: 45, Hazel: null (*)

Dick: 0, Hazel: 0

Both queries will fail because the syntax of DBMS_OUTPUT.PUT_LINE is incorrect

Correct

10. When a user session changes the value of a package variable, the new value can immediately be seen by other sessions. True or False? Mark for Review
(1) Points

True

False (*)

Incorrect. Refer to Section 9.

Test: Mid Term Exam Semester 2

Review your answers, feedback, and question scores below. An asterisk (*) indicates a correct answer.

Section 9

11. Package OLDPACK is in your schema. What will happen when the following statement is executed?
DROP PACKAGE oldpack;
Mark for Review
(1) Points

The body will be dropped but the specification will be retained.

The specification will be dropped but the body will be retained.

Both the specification and the body will be dropped. (*)

The statement will fail because you must drop the body before you can drop the specification.

Incorrect. Refer to Section 9.

12. In a package, public components are declared in the specification but private components are not. True or False? Mark for Review
(1) Points

True (*)

False

Correct

13. A package contains both public and private subprograms. Which one of the following statements is true? Mark for Review
(1) Points

Each subprogram is loaded into memory when it is first invoked.

The public subprograms are all loaded into memory at the same time, but the private subprograms are loaded into memory one at a time as they are invoked.

The whole package is loaded into memory when the first call is made to any subprogram in the package. (*)

If three users invoke three different subprograms in the package, there will be three copies of the code in memory.

Incorrect. Refer to Section 9.

14. Package NEWPACK contains several procedures and functions, including private function PRIVFUNC. From where can PRIVFUNC be invoked? (Choose two.) Mark for Review
(1) Points

(Choose all correct answers)

From an anonymous block

From any procedure in NEWPACK (*)

From any private function in another package

From any function in NEWPACK (*)

From any public procedure in another package

Incorrect. Refer to Section 9.

15. We want to remove the specification (but not the body) of package BIGPACK from the database. Which of the following commands will do this? Mark for Review
(1) Points

DROP PACKAGE bigpack;

DROP PACKAGE SPECIFICATION bigpack;

DROP PACKAGE bigpack SPECIFICATION;

DROP PACKAGE HEADER bigpack;

None of the above (*)

Incorrect. Refer to Section 9.

16. Which of the following will display the detailed code of the subprograms in package DEPTPACK in your schema ? Mark for Review
(1) Points

```
SELECT text FROM USER_SOURCE
WHERE name = 'DEPTPACK'
AND type = 'PACKAGE'
ORDER BY line;
```

```
SELECT text FROM USER_SOURCE
WHERE name = 'DEPTPACK'
AND type = 'PACKAGE BODY'
ORDER BY line;
```
(*)

```
SELECT text FROM USER_SOURCE
WHERE object_name = 'DEPTPACK'
AND object_type = 'PACKAGE BODY'
ORDER BY line;
```

```
SELECT text FROM USER_SOURCE
WHERE name = 'DEPTPACK'
AND type = 'BODY'
ORDER BY line;
```

Incorrect. Refer to Section 9.

17. When a change is made to the detailed code of a public procedure in a package (but not to the procedure's name or parameters), both the specification and the body must be recompiled. True or False? Mark for Review
(1) Points

True

False (*)

Incorrect. Refer to Section 9.

18. What is wrong with the following syntax for creating a package specification?

```
CREATE OR REPLACE PACKAGE mypack IS
    g_constant1 NUMBER(6) := 100;
    FUNCTION func1 (p_param1 IN VARCHAR2);
    FUNCTION func2;
END mypack;
```

Mark for Review
(1) Points

You cannot declare constants in the specification.

A package must contain at least one procedure.

The RETURN datatype of the functions must be specified.
(*)

The first line should be:
CREATE OR REPLACE PACKAGE SPECIFICATION mypack IS

Nothing is wrong, this code contains no errors.

Incorrect. Refer to Section 9.

19. The following package specification has been created:

```
CREATE OR REPLACE PACKAGE mypack IS
    FUNCTION myfunc(p_funcparam DATE) RETURN BOOLEAN;
    PROCEDURE myproc(p_procparam IN NUMBER);
END mypack;
```

Which of the following will correctly invoke the package subprograms? (Choose two.)
Mark for Review
(1) Points

(Choose all correct answers)

mypack.myfunc('22-JAN-07');

mypack.myproc(35);
(*)

IF NOT mypack.myfunc(SYSDATE) THEN
DBMS_OUTPUT.PUT_LINE('Message');
END IF;

(*)

myproc(40);

v_num := mypack.myproc(22);

Incorrect. Refer to Section 9.

20. Which part of a package must be created first, the specification or the body? Mark for Review
(1) Points

The body

The specification (*)

The specification and body must be created at the same time.

It does not matter which is created first.

The body can be created first, but only if the package has no specification.

Correct

Test: Mid Term Exam Semester 2

Review your answers, feedback, and question scores below. An asterisk (*) indicates a correct answer.

Section 9

21. Every subprogram which has been declared in a package specification must also be included in the package body. Triue or False? Mark for Review
(1) Points

True (*)

False

Incorrect. Refer to Section 9.

22. The UTL_FILE package can be used to create binary files such as JPEGs as well as text files. True or False? Mark for Review
(1) Points

True

False (*)

Incorrect. Refer to Section 9.

23. Why is it better to use DBMS_OUTPUT only in anonymous blocks, not inside stored subprograms such as procedures? Mark for Review
(1) Points

Because DBMS_OUTPUT cannot be used inside procedures

Because anonymous blocks display messages while the block is executing, while procedures do not display anything until their execution has finished

Because DBMS_OUTPUT should be used only for testing and debugging PL/SQL code (*)

Because DBMS_OUTPUT can raise a NO_DATA_FOUND exception if used inside a packaged procedure

Incorrect. Refer to Section 9.

24. The DBMS_OUTPUT.PUT procedure places text in a buffer but does not display the contents of the buffer. True or False? Mark for Review
(1) Points

True (*)

False

Incorrect. Refer to Section 9.

25. DBMS_OUTPUT.PUT_LINE can be invoked from inside a private packaged function. True or False? Mark for Review
(1) Points

True (*)

False

Correct

Section 10

26. What type of database object would you create to write an auditing record automatically every time a user connects to the database? Mark for Review
(1) Points

A procedure

A complex view

A trigger (*)

A function

A package

Incorrect. Refer to Section 10.

27. Which of the following are good guidelines to follow when creating a database trigger? (Choose two.) Mark for Review
(1) Points

(Choose all correct answers)

Where possible, use a trigger to enforce a foreign key constraint.

Use triggers to override privilege checking and view other users' private tables.

Do not use a trigger to replace or duplicate something which the Oracle Server does automatically. (*)

Use triggers to prevent unauthorized users from SELECTing confidential data.

Do not create a trigger that automatically fires another trigger. (*)

Incorrect. Refer to Section 10.

28. A trigger can be created in the database or within an application. True or False? Mark for Review
(1) Points

True (*)

False

Correct

29. The following objects have been created in a user's schema:
- a function FUNC1
- A package PACK1 which contains a public procedure PACKPROC and a private function PACKFUNC
- a trigger TRIGG1.
The procedure and functions each accept a single IN parameter of type NUMBER, and the functions return BOOLEANs. Which of the following calls to these objects (from an anonymous block) are correct? (Choose two.) Mark for Review
(1) Points

(Choose all correct answers)

pack1.packproc(25); (*)

SELECT func1(100) FROM dual;

trigg1;

IF pack1.packfunc(40) THEN ...

IF func1(75) THEN ... (*)

Incorrect. Refer to Section 10.

30. A business rule states that an employee's salary cannot be greater than 99,999.99 or less than 0. The best way to enforce this rule is by using: Mark for Review
(1) Points

A datatype of NUMBER(7,2) for the SALARY column

A database trigger

A check constraint (*)

An application trigger

A view

Incorrect. Refer to Section 10.

Test: Mid Term Exam Semester 2

Review your answers, feedback, and question scores below. An asterisk (*) indicates a correct answer.

Section 10

31. You can use a trigger to prevent rows from being deleted from the EMPLOYEES table on Mondays. True or False? Mark for Review
(1) Points

True (*)

False

Incorrect. Refer to Section 10.

32. Which of the following are NOT stored inside the database? (Choose two.) Mark for Review
(1) Points

(Choose all correct answers)

A PL/SQL package specification

A database trigger

An anonymous block (*)

An application trigger (*)

A sequence

Incorrect. Refer to Section 10.

33. Examine the following code. To create a row trigger, what code should be included at Line A?

```
CREATE TRIGGER dept_trigg
AFTER UPDATE OR DELETE ON departments
    -- Line A
BEGIN ...
```

Mark for Review
(1) Points

AFTER EACH ROW

FOR EVERY ROW

FOR EACH ROW (*)

ON EACH ROW

ON EVERY ROW

Incorrect. Refer to Section 10.

34. In the following code:

```
CREATE TRIGGER mytrigg
INSTEAD OF INSERT OR UPDATE ON my_object_name
FOR EACH ROW
BEGIN ...
```

my_object_name can be the name of a table. True or False?
Mark for Review
(1) Points

True

False (*)

Incorrect. Refer to Section 10.

35. Examine the following trigger. It should raise an application error if a user tries to update an employee's last name. It should allow updates to all other columns of the EMPLOYEES table. What should be coded at line A?

```
CREATE TRIGGER stop_ln_trigg
BEFORE UPDATE ON employees
BEGIN
      -- Line A
    RAISE_APPLICATION_ERROR(-20201,'Updating last name not allowed');
    END IF;
```

```
END;
```

Mark for Review
(1) Points

IF UPDATING LAST_NAME THEN

IF UPDATING('LAST_NAME') THEN (*)

IF UPDATE('LAST_NAME') THEN

IF UPDATING THEN

Incorrect. Refer to Section 10.

36. There are 3 employees in department 90 and 5 employees in department 50. The following trigger has been created:

```
CREATE TRIGGER upd_emp_trigg
AFTER UPDATE ON employees
FOR EACH ROW
BEGIN
...
```

A user now executes:

```
UPDATE employees SET department_id = 50
    WHERE department_id = 90;
```

How many times will the trigger fire?
Mark for Review
(1) Points

Once

Three times (*)

Four times

Five times

Eight times

Incorrect. Refer to Section 10.

37. After the following SQL statement is executed, all the triggers on the DEPARTMENTS table will no longer fire, but will remain in the database. True or False?

```
ALTER TABLE departments DISABLE ALL TRIGGERS;
```

Mark for Review
(1) Points

True (*)

False

Incorrect. Refer to Section 10.

38. Which of the following will remove a trigger in your schema named EMP_TRIGG from the database? Mark for Review
(1) Points

DROP emp_trigg TRIGGER;

ALTER TRIGGER emp_trigg DISABLE;

DROP TRIGGER emp_trigg; (*)

REMOVE TRIGGER emp_trigg;

None of the above

Correct

39. You need to disable all triggers that are associated with DML statements on the DEPARTMENTS table. Which of the following commands should you use? Mark for Review
(1) Points

ALTER TABLE departments DISABLE ALL TRIGGERS; (*)

ALTER TRIGGER DISABLE ALL ON departments;

ALTER TABLE departments DISABLE TRIGGERS;

DISABLE ALL TRIGGERS ON departments;

ALTER TABLE departments DROP ALL TRIGGERS;

Correct

40. You need to create a trigger that will fire whenever an employee's salary or job_id is updated, but not when any other column of the EMPLOYEES table is updated. Which of the following is the correct syntax to do this? Mark for Review
(1) Points

CREATE TRIGGER emp_upd_trigg
AFTER UPDATE ON employees (salary, job_id)
BEGIN ...

CREATE TRIGGER emp_upd_trigg
AFTER UPDATE OF salary, job_id ON employees
BEGIN ...
(*)

CREATE TRIGGER emp_upd_trigg
AFTER UPDATE OF (salary, job_id) ON employees
BEGIN ...

CREATE TRIGGER emp_upd_trigg
AFTER UPDATE OF salary OR job_id ON employees
BEGIN ...

Incorrect. Refer to Section 10.

Page 4 of 5

Test: Mid Term Exam Semester 2

Review your answers, feedback, and question scores below. An asterisk (*) indicates a correct answer.

Section 10

41. A DML statement trigger fires only once for each triggering DML statement, while a row trigger fires once for each row processed by the triggering statement. True or False? Mark for Review
(1) Points

True (*)

False

Incorrect. Refer to Section 10.

42. What is wrong with the following code?

```
CREATE TRIGGER dept_trigg
BEFORE UPDATE OF department_name ON departments
BEGIN
    DBMS_OUTPUT.PUT_LINE(:NEW.department_name);
END;
```

Mark for Review
(1) Points

You cannot use :NEW in a BEFORE trigger, only in an AFTER trigger.

You cannot use :NEW or :OLD in a statement trigger. (*)

You cannot use DBMS_OUTPUT.PUT_LINE inside a trigger.

The second line should be:

```
BEFORE UPDATE ON departments.department_name
```

Incorrect. Refer to Section 10.

43. What is wrong with the following code?

```
CREATE OR REPLACE TRIGGER emp_dept_trigg
BEFORE UPDATE OR DELETE ON employees, departments
BEGIN
    ...
```

Mark for Review
(1) Points

One trigger can be associated with only one table
(*)

The second line should be:
BEFORE (UPDATE,DELETE) ON employees, departments

DML triggers must be row triggers, so FOR EACH ROW is missing

The second line should be:
BEFORE UPDATE OR DELETE ON employees OR departments

Incorrect. Refer to Section 10.

44. Which of the following statements could cause a DDL trigger to fire? Mark for Review
(1) Points

DROP TABLE employees;

ALTER TABLE departments ADD (budget NUMBER(8,2));

CREATE TABLE newemp AS SELECT * FROM employees;

TRUNCATE TABLE locations;

All of the above (*)

Incorrect. Refer to Section 10.

45. You want to prevent any objects in your schema from being altered or dropped. You decide to create the following trigger:

```
CREATE TRIGGER stop_ad_trigg
    -- Line A
BEGIN
    RAISE_APPLICATION_ERROR(-20203,'Invalid Operation');
END;
```

What should you code at Line A ?
Mark for Review
(1) Points

AFTER ALTER OR DROP ON SCHEMA

INSTEAD OF ALTER OR DROP ON SCHEMA

BEFORE ALTER OR DROP ON SCHEMA (*)

BEFORE ALTER, DROP ON SCHEMA

AFTER ALTER, DROP ON SCHEMA

Correct

46. Examine the following code:

```
CREATE TRIGGER emp_trigg
AFTER UPDATE OF salary ON employees
FOR EACH ROW
DECLARE
    v_count NUMBER;
BEGIN
    -- Line A
END;
```

Which of the following statements is NOT allowed at Line A?
Mark for Review
(1) Points

SELECT count(*) INTO v_count FROM departments;

UPDATE employees SET job_id = 'IT_PROG' WHERE employee_id = :OLD.employee_id;

SELECT count(*) INTO v_count FROM employees; (*)

DBMS_OUTPUT.PUT_LINE('A salary was updated');

None. All of the above are allowed.

Incorrect. Refer to Section 10.

47. Examine this code:

```
CREATE TRIGGER de_trigg
    -- Line A
BEGIN ...
```

Which of the following are NOT valid at Line A ? (Choose two.)
Mark for Review
(1) Points

(Choose all correct answers)

AFTER LOGOFF ON SCHEMA (*)

AFTER LOGON ON SCHEMA

BEFORE LOGOFF ON SCHEMA

BEFORE DISCONNECT ON SCHEMA (*)

AFTER SERVERERROR ON SCHEMA

Incorrect. Refer to Section 10.

48. Which kinds of trigger can cause a mutating table problem? (Choose two.)
Mark for Review
(1) Points

(Choose all correct answers)

BEFORE UPDATE row triggers (*)

DDL triggers

AFTER DELETE row triggers (*)

Database Event triggers

INSTEAD OF triggers

Incorrect. Refer to Section 10.

49. What is wrong with the following code?

```
CREATE TRIGGER call_trigg
AFTER LOGOFF ON SCHEMA
BEGIN
    CALL drop_proc;
END;
```

Mark for Review
(1) Points

You cannot code an AFTER LOGOFF trigger

When using CALL, you must not code BEGIN

When using CALL, you must not code END;

The CALL statement must not end with a semicolon (;)

All of the above (*)

Incorrect. Refer to Section 10.

50. Examine this code:

```
CREATE TRIGGER new_trigg
AFTER CREATE ON reserved_word
BEGIN ...
```

Which of the following can be used in place of reserved_word? (Choose two.)
Mark for Review
(1) Points

(Choose all correct answers)

TABLE

SCHEMA (*)

USER

DATABASE (*)

TABLE employees

Incorrect. Refer to Section 10.

1. Why is it better to use DBMS_OUTPUT only in anonymous blocks, not inside stored subprograms such as procedures? Mark for Review
(1) Points

Because DBMS_OUTPUT cannot be used inside procedures

Because anonymous blocks display messages while the block is executing, while procedures do not display anything until their execution has finished

Because DBMS_OUTPUT should be used only for testing and debugging PL/SQL code (*)

Because DBMS_OUTPUT can raise a NO_DATA_FOUND exception if used inside a packaged procedure

Correct

2. DBMS_OUTPUT.PUT_LINE can be invoked from inside a private packaged function. True or False? Mark for Review
(1) Points

True (*)

False

Correct

3. Which of the following exceptions can be raised ONLY when using the UTL_FILE package? (Choose two) Mark for Review
(1) Points

(Choose all correct answers)

INVALID_PATH (*)

NO_DATA_FOUND

VALUE_ERROR

READ_ERROR (*)

E_MYEXCEP

Correct

4. What will be displayed when the following code is executed?

```
BEGIN
    DBMS_OUTPUT.PUT('I do like');
    DBMS_OUTPUT.PUT_LINE('to be');
```

```
    DBMS_OUTPUT.PUT('beside the seaside');
END;
```

Mark for Review
(1) Points

I do like to be
beside the seaside

I do like
to be
beside the seaside

I do like to be

I do liketo be
(*)

I do like to be beside the seaside

Incorrect. Refer to Section 9.

5. A public function in a package is invoked from within a SQL statement. The function's code can include a COMMIT statement. True or False? Mark for Review
(1) Points

True

False (*)

Correct

6. Examine the following code:

```
CREATE OR REPLACE PACKAGE emppack IS
    PROCEDURE upd_emp (p_empno IN NUMBER, p_salary IN NUMBER);
END emppack;
CREATE OR REPLACE PACKAGE BODY emppack IS
    -- Line A
    PROCEDURE upd_emp (p_empno IN NUMBER, p_salary IN NUMBER) IS
       BEGIN
          IF NOT sal_ok(p_salary) THEN
             RAISE_APPLICATION_ERROR(-20201,'Invalid salary');
          END IF;
    END upd_emp;
    FUNCTION sal_ok(pf_salary NUMBER) RETURN BOOLEAN IS
       BEGIN
          IF pf_salary > 50000 THEN RETURN FALSE;
          ELSE RETURN TRUE;
          END IF;
    END sal_ok;
END emppack;
```

What must be coded at Line A for this package to compile successfully?
Mark for Review

(1) Points

FUNCTION sal_ok;

FUNCTION sal_ok(pf_salary NUMBER);

FUNCTION sal_ok(pf_salary NUMBER) RETURN BOOLEAN; (*)

PROCEDURE upd_emp (p_empno IN NUMBER, p_salary IN NUMBER);

Nothing is needed at Line A

Incorrect. Refer to Section 9.

7. Package HRPACK contains the following public function:

```
FUNCTION empfunc (p_deptno NUMBER) RETURN NUMBER IS
BEGIN
    UPDATE employees SET salary = salary * 1.1
       WHERE department_id = p_deptno;
    RETURN SQL%ROWCOUNT;
END empfunc;
```

What will happen when the following SQL statement is executed?

```
SELECT department_name, hrpack.empfunc(department_id)
FROM departments;
```

Mark for Review

(1) Points

The SELECT will fail because you cannot return SQL%ROWCOUNT from a packaged function.

The SELECT will fail because you cannot call packaged functions from within a SQL statement.

The SELECT will fail because you cannot execute a DML statement from within a query.

The SELECT will succeed because it is referencing a different table from the function. (*)

Correct

8. Which two of these declarations cannot be in the same package specification?

```
PROCEDURE myproc (p1 NUMBER, p2 VARCHAR2);
PROCEDURE myproc (p1 VARCHAR2, p2 NUMBER);
PROCEDURE myproc (p1 NUMBER, p2 CHAR);
PROCEDURE myproc (p1 NUMBER);
```

Mark for Review

(1) Points

1 and 2

1 and 3 (*)

2 and 3

3 and 4

1 and 4

Correct

9. A cursor is declared in a package specification. User SIOBHAN opens the cursor and fetches the first three rows from the cursor's active set, but does not close the cursor.
User FRED now connects to the database. FRED can immediately fetch the next three rows without opening the cursor. True or False? Mark for Review
(1) Points

True

False (*)

Correct

10. Package MULTIPACK declares the following global variable:

```
g_myvar NUMBER;
```

User DICK executes the following:

```
multipack.g_myvar := 45;
```

User HAZEL now connects to the database. Both users immediately execute:

```
BEGIN
    DBMS_OUTPUT.PUT_LINE(multipack.g_myvar);
END;
```

What values will Dick and Hazel see?
Mark for Review
(1) Points

Dick: 45, Hazel: 45

Dick: 45, Hazel: 0

Dick: 45, Hazel: null (*)

Dick: 0, Hazel: 0

Both queries will fail because the syntax of DBMS_OUTPUT.PUT_LINE is incorrect

Incorrect. Refer to Section 9.

11. We want to remove the specification (but not the body) of package BIGPACK from the database. Which of the following commands will do this? Mark for Review
(1) Points

DROP PACKAGE bigpack;

DROP PACKAGE SPECIFICATION bigpack;

DROP PACKAGE bigpack SPECIFICATION;

DROP PACKAGE HEADER bigpack;

None of the above (*)

Correct

12. Which of the following will display the detailed code of the subprograms in package DEPTPACK in your schema ? Mark for Review
(1) Points

```
    SELECT text FROM USER_SOURCE
WHERE name = 'DEPTPACK'
AND type = 'PACKAGE'
ORDER BY line;
```

```
    SELECT text FROM USER_SOURCE
WHERE name = 'DEPTPACK'
AND type = 'PACKAGE BODY'
ORDER BY line;
```
(*)

```
    SELECT text FROM USER_SOURCE
WHERE object_name = 'DEPTPACK'
AND object_type = 'PACKAGE BODY'
ORDER BY line;
```

```
    SELECT text FROM USER_SOURCE
WHERE name = 'DEPTPACK'
AND type = 'BODY'
ORDER BY line;
```

Incorrect. Refer to Section 9.

13. Examine the following package specification:

```
CREATE OR REPLACE PACKAGE taxpack IS
    CURSOR empcurs IS SELECT * FROM employees;
    PROCEDURE taxproc;
END mypack;
```

The package body of TAXPACK also includes a function called TAXFUNC. Which one of the following statements is NOT true?
Mark for Review
(1) Points

The procedure can be invoked by:

```
BEGIN
    taxpack.taxproc;
END;
```

The packaage will not compile because you cannot declare a cursor in the specification.
(*)

TAXPROC is a public procedure and TAXFUNC is a private function

TAXPROC can invoke TAXFUNC if TAXPROC is coded before TAXFUNC

TAXPROC can open the cursor

Correct

14. When a change is made to the detailed code of a public procedure in a package (but not to the procedure's name or parameters), both the specification and the body must be recompiled. True or False? Mark for Review
(1) Points

True

False (*)

Correct

15. Package OLDPACK is in your schema. What will happen when the following statement is executed?
DROP PACKAGE oldpack;
Mark for Review
(1) Points

The body will be dropped but the specification will be retained.

The specification will be dropped but the body will be retained.

Both the specification and the body will be dropped. (*)

The statement will fail because you must drop the body before you can drop the specification.

Correct

16. We need to declare a package variable named MYVAR, which can be referenced by any subprogram in the package but can NOT be referenced from outside the package. In the following code, where should MYVAR be declared?

```
CREATE OR REPLACE PACKAGE varpack IS
    -- Point A
...
END varpack;
CREATE OR REPLACE PACKAGE BODY varpack IS
    -- Point B
PROCEDURE varproc IS
    -- Point C
    BEGIN
       ...
    END varproc;
PROCEDURE ...
...
    -- Point D
END varpack;
```

Mark for Review
(1) Points

Point A

Point B (*)

Point C

Point D

Point B or Point C, they will both work

Incorrect. Refer to Section 9.

17. In a package, public components are declared in the specification but private components are not. True or False? Mark for Review
(1) Points

True (*)

False

Correct

18. Which one of the following can NOT be part of a Package ? Mark for Review
(1) Points

Procedures

Explicit cursors

Triggers (*)

Functions

Global variables

Correct

19. Which of the following are good reasons for creating and using Packages?

Related procedures, functions and variables can be grouped together as a single unit

We can recompile the package body without having to recompile the specification
We can create packages without needing any system privileges
We can declare INDEX BY tables and use them as parameters
Mark for Review
(1) Points

A and B

A, B and C

A and C

A, B and D (*)

A, B, C and D

Incorrect. Refer to Section 9.

20. What is wrong with the following syntax for creating a package specification?

```
CREATE OR REPLACE PACKAGE mypack IS
    g_constant1 NUMBER(6) := 100;
    FUNCTION func1 (p_param1 IN VARCHAR2);
    FUNCTION func2;
END mypack;
```

Mark for Review
(1) Points

You cannot declare constants in the specification.

A package must contain at least one procedure.

The RETURN datatype of the functions must be specified.
(*)

The first line should be:
CREATE OR REPLACE PACKAGE SPECIFICATION mypack IS

Nothing is wrong, this code contains no errors.

Incorrect. Refer to Section 9.

21. Which part of a package must be created first, the specification or the body?
Mark for Review
(1) Points

The body

The specification (*)

The specification and body must be created at the same time.

It does not matter which is created first.

The body can be created first, but only if the package has no specification.

Correct

22. Examine the following code:

```
CREATE OR REPLACE PROCEDURE myproc IS
    CURSOR c_curs IS SELECT view_name FROM user_views;
BEGIN
    FOR v_curs_rec IN c_curs LOOP
       EXECUTE IMMEDIATE 'DROP VIEW ' || v_curs_rec.view_name;
    END LOOP;
END;
```

What will happen when this procedure is invoked?
Mark for Review

(1) Points

All views in the user's schema will be dropped. (*)

The procedure will not compile successfully because the syntax of EXECUTE IMMEDIATE is incorrect.

The procedure will raise an exception because Dynamic SQL can drop tables but cannot drop views.

The procedure will raise an exception because one of the views is a complex view.

Correct

23. The easiest way to include DDL statements in a PL/SQL block is to use the DBMS_SQL package. True or False? Mark for Review

(1) Points

True

False (*)

Correct

24. A SQL statement can pass through several stages. Which of the following is NOT one of these stages? Mark for Review

(1) Points

BIND

FETCH

PARSE

RETURN (*)

EXECUTE

Correct

25. MARY wants HENRY to be able to query her EMPLOYEES table. Mary executes the following code:

```
DECLARE
    v_grant_stmt VARCHAR2(50);
BEGIN
    v_grant_stmt := 'GRANT SELECT ON employees TO henry';
    DBMS_SQL.EXECUTE(v_grant_stmt);
END;
```

Mary has successfully granted the privilege to Henry. True or False? Mark for Review

(1) Points

True

False (*)

Correct

Section 10

26. What type of database object would you create to write an auditing record automatically every time a user connects to the database? Mark for Review
(1) Points

A procedure

A complex view

A trigger (*)

A function

A package

Correct

27. A trigger can be created in the database or within an application. True or False? Mark for Review
(1) Points

True (*)

False

Correct

28. A trigger can be a public subprogram within a PL/SQL package. True or False? Mark for Review
(1) Points

True

False (*)

Correct

29. The following objects have been created in a user's schema:
- a function FUNC1
- A package PACK1 which contains a public procedure PACKPROC and a private function PACKFUNC
- a trigger TRIGG1.
The procedure and functions each accept a single IN parameter of type NUMBER, and the functions return BOOLEANs. Which of the following calls to these objects (from an anonymous block) are correct? (Choose two.) Mark for Review
(1) Points

(Choose all correct answers)

pack1.packproc(25); (*)

SELECT func1(100) FROM dual;

trigg1;

IF pack1.packfunc(40) THEN ...

IF func1(75) THEN ... (*)

Incorrect. Refer to Section 10.

30. A business rule states that an employee's salary cannot be greater than 99,999.99 or less than 0. The best way to enforce this rule is by using: Mark for Review
(1) Points

A datatype of NUMBER(7,2) for the SALARY column

A database trigger

A check constraint (*)

An application trigger

A view

Incorrect. Refer to Section 10.

31. You can code COMMIT and ROLLBACK statements in a trigger body. True or False? Mark for Review
(1) Points

True

False (*)

Correct

32. Which of the following are NOT stored inside the database? (Choose two.) Mark for Review
(1) Points

(Choose all correct answers)

A PL/SQL package specification

A database trigger

An anonymous block (*)

An application trigger (*)

A sequence

Incorrect. Refer to Section 10.

33. Which dictionary view shows the detailed code of a trigger body? Mark for Review
(1) Points

USER_SOURCE

USER_TRIGGERS (*)

USER_OBJECTS

USER_DML_TRIGGERS

USER_SUBPROGRAMS

Correct

34. MARY and JOE's schemas each contain an EMPLOYEES table. JOE creates the following trigger:

```
CREATE TRIGGER upd_trigg
AFTER DELETE ON joe.employees
FOR EACH ROW
BEGIN
    DELETE FROM mary.employees
       WHERE employee_id = :OLD.employee_id;
END;
```

A third user TOM needs to delete rows from JOE's EMPLOYEES table. What object privileges will TOM and JOE need?
Mark for Review
(1) Points

TOM does not need any object privileges, but JOE needs DELETE on both TOM.EMPLOYEES and MARY.EMPLOYEES

TOM needs DELETE on JOE.EMPLOYEES and JOE needs DELETE on MARY.EMPLOYEES (*)

JOE does not need any object privileges, but TOM needs DELETE on MARY.EMPLOYEES

TOM needs DELETE on MARY.EMPLOYEES and JOE needs EXECUTE on TOM.UPD_TRIGG

Correct

35. After the following SQL statement is executed, all the triggers on the DEPARTMENTS table will no longer fire, but will remain in the database. True or False?

```
ALTER TABLE departments DISABLE ALL TRIGGERS;
```

Mark for Review
(1) Points

True (*)

False

Correct

36. What is wrong with the following code?

```
CREATE TRIGGER call_trigg
AFTER LOGOFF ON SCHEMA
BEGIN
    CALL drop_proc;
END;
```

Mark for Review
(1) Points

You cannot code an AFTER LOGOFF trigger

When using CALL, you must not code BEGIN

When using CALL, you must not code END;

The CALL statement must not end with a semicolon (;)

All of the above (*)

Incorrect. Refer to Section 10.

37. Which kinds of trigger can cause a mutating table problem? (Choose two.)
Mark for Review
(1) Points

(Choose all correct answers)

BEFORE UPDATE row triggers (*)

DDL triggers

AFTER DELETE row triggers (*)

Database Event triggers

INSTEAD OF triggers

Incorrect. Refer to Section 10.

38. A trigger automatically inserts a row into a logging table every time a user's session receives this error message:
ORA-00942: table or view does not exist
What kind of trigger is this? Mark for Review
(1) Points

A row trigger

A statement trigger

A database event trigger (*)

A DDL trigger

An AFTER trigger

Incorrect. Refer to Section 10.

39. The database administrator wants to write a log record every time an Oracle Server error occurs in any user's session. The DBA creates the following trigger:

```
CREATE TRIGGER log_errs_trigg
    -- Line A
BEGIN
    INSERT INTO errlog_table VALUES (...);
END;
```

What should the DBA code at Line A ?

Mark for Review
(1) Points

AFTER ERROR ON DATABASE

AFTER SERVER ERROR ON DATABASE

AFTER SERVERERROR ON SCHEMA

AFTER SERVERERROR ON DATABASE (*)

AFTER ORACLE ERROR ON SCHEMA

Incorrect. Refer to Section 10.

40. Examine this code:

```
CREATE TRIGGER new_trigg
AFTER CREATE ON reserved_word
BEGIN ...
```

Which of the following can be used in place of reserved_word? (Choose two.)

Mark for Review
(1) Points

(Choose all correct answers)

TABLE

SCHEMA (*)

USER

DATABASE (*)

TABLE employees

Correct

41. What is the purpose of using the CALL statement in a trigger? Mark for Review
(1) Points

It allows an INSTEAD OF trigger to be a statement trigger.

It allows the trigger body code to be placed in a separate procedure. (*)

It prevents cascading triggers.

It allows the trigger body code to be placed in a separate procedure or function.

It allows both DML events and DDL events to be handled using a single trigger.

Incorrect. Refer to Section 10.

42. Examine the following code:

```
CREATE TRIGGER emp_trigg
AFTER UPDATE OF salary ON employees
FOR EACH ROW
DECLARE
    v_count NUMBER;
BEGIN
    -- Line A
END;
```

Which of the following statements is NOT allowed at Line A?
Mark for Review
(1) Points

SELECT count(*) INTO v_count FROM departments;

UPDATE employees SET job_id = 'IT_PROG' WHERE employee_id = :OLD.employee_id;

SELECT count(*) INTO v_count FROM employees; (*)

DBMS_OUTPUT.PUT_LINE('A salary was updated');

None. All of the above are allowed.

Incorrect. Refer to Section 10.

43. What is wrong with the following code?

```
CREATE TRIGGER dept_trigg
BEFORE UPDATE OF department_name ON departments
BEGIN
    DBMS_OUTPUT.PUT_LINE(:NEW.department_name);
END;
```

Mark for Review
(1) Points

You cannot use :NEW in a BEFORE trigger, only in an AFTER trigger.

You cannot use :NEW or :OLD in a statement trigger. (*)

You cannot use DBMS_OUTPUT.PUT_LINE inside a trigger.

The second line should be:

```
BEFORE UPDATE ON departments.department_name
```

Incorrect. Refer to Section 10.

44. There are five employees in department 50. The following trigger is created:

```
CREATE TRIGGER upd_emp
AFTER UPDATE ON employees
BEGIN
    INSERT INTO audit_table VALUES (USER, SYSDATE);
END;
```

A user now executes:

```
UPDATE employees SET salary = salary * 1.1
    WHERE department_id = 50;
```

How many rows will be inserted into audit_table?

Mark for Review
(1) Points

One (*)

Two

Five

Six

None of the above

Correct

45. Examine the following code:

```
CREATE TRIGGER emp_trigg
    -- Line A
BEGIN
    INSERT INTO log_table VALUES (USER, SYSDATE);
END;
```

Which of the following can NOT be coded at Line A?

Mark for Review
(1) Points

BEFORE UPDATE ON employees

AFTER INSERT OR DELETE ON employees

AFTER SELECT ON employees (*)

BEFORE DELETE ON employees

AFTER UPDATE OF last_name ON employees

Incorrect. Refer to Section 10.

46. The following code will successfully create emp_trigg: True or False?

```
CREATE OR REPLACE TRIGGER emp_trigg
BEFORE DELETE OF salary ON employees
BEGIN
    RAISE_APPLICATION_ERROR(-20202,'Deleting salary is not allowed');
END;
```

Mark for Review
(1) Points

True

False (*)

Incorrect. Refer to Section 10.

47. INSTEAD OF triggers are always row triggers, even if FOR EACH ROW is omitted. True or False? Mark for Review
(1) Points

True (*)

False

Incorrect. Refer to Section 10.

48. Examine the following code. To create a row trigger, what code should be included at Line A?

```
CREATE TRIGGER dept_trigg
AFTER UPDATE OR DELETE ON departments
    -- Line A
BEGIN ...
```

Mark for Review
(1) Points

AFTER EACH ROW

FOR EVERY ROW

FOR EACH ROW (*)

ON EACH ROW

ON EVERY ROW

Correct

49. There are 3 employees in department 90 and 5 employees in department 50. The following trigger has been created:

```
CREATE TRIGGER upd_emp_trigg
AFTER UPDATE ON employees
FOR EACH ROW
BEGIN
...
```

A user now executes:

```
UPDATE employees SET department_id = 50
    WHERE department_id = 90;
```

How many times will the trigger fire?
Mark for Review
(1) Points

Once

Three times (*)

Four times

Five times

Eight times

Correct

50. In the following code:

```
CREATE TRIGGER mytrigg
INSTEAD OF INSERT OR UPDATE ON my_object_name
FOR EACH ROW
BEGIN ...
```

my_object_name can be the name of a table. True or False?
Mark for Review
(1) Points

True

False (*)

Correct

Test: Mid Term Exam Semester 2

Review your answers, feedback, and question scores below. An asterisk (*) indicates a correct answer.

Section 9

1. Every subprogram which has been declared in a package specification must also be included in the package body. Triue or False? Mark for Review
(1) Points

True (*)

False

Correct

2. The following package specification has been created:

```
CREATE OR REPLACE PACKAGE mypack IS
    FUNCTION myfunc(p_funcparam DATE) RETURN BOOLEAN;
    PROCEDURE myproc(p_procparam IN NUMBER);
END mypack;
```

Which of the following will correctly invoke the package subprograms? (Choose two.)
Mark for Review
(1) Points

(Choose all correct answers)

mypack.myfunc('22-JAN-07');

mypack.myproc(35);
(*)

```
    IF NOT mypack.myfunc(SYSDATE) THEN
    DBMS_OUTPUT.PUT_LINE('Message');
END IF;
(*)


    myproc(40);


    v_num := mypack.myproc(22);
```

Incorrect. Refer to Section 9.

3. Package MYPACK contains procedure MYPROC. You can see which parameters MYPROC uses by executing: DESCRIBE mypack.myproc. True or False? Mark for Review
(1) Points

True

False (*)

Incorrect. Refer to Section 9.

4. Which part of a package must be created first, the specification or the body? Mark for Review
(1) Points

The body

The specification (*)

The specification and body must be created at the same time.

It does not matter which is created first.

The body can be created first, but only if the package has no specification.

Incorrect. Refer to Section 9.

5. You want to create a function which drops a table. You write the following code:

```
CREATE OR REPLACE FUNCTION droptab
    (p_tab_name IN VARCHAR2)
RETURN BOOLEAN IS
BEGIN
    DROP TABLE p_tab_name;
    RETURN TRUE;
EXCEPTION
    WHEN OTHERS THEN RETURN FALSE;
END;
```

Why will this procedure not compile successfully?
Mark for Review
(1) Points

Because you can never drop a table from inside a function

Because the PL/SQL compiler cannot check if the argument of p_tab_name is a valid table-name (*)

Because you do not have the privilege needed to drop a table

Because you cannot use RETURN in the exception section

Incorrect. Refer to Section 9.

6. A SQL statement can pass through several stages. Which of the following is NOT one of these stages? Mark for Review
(1) Points

BIND

FETCH

PARSE

RETURN (*)

EXECUTE

Incorrect. Refer to Section 9.

7. Examine the following code:

```
CREATE OR REPLACE PROCEDURE myproc IS
    CURSOR c_curs IS SELECT view_name FROM user_views;
BEGIN
    FOR v_curs_rec IN c_curs LOOP
       EXECUTE IMMEDIATE 'DROP VIEW ' || v_curs_rec.view_name;
    END LOOP;
END;
```

What will happen when this procedure is invoked?
Mark for Review
(1) Points

All views in the user's schema will be dropped. (*)

The procedure will not compile successfully because the syntax of EXECUTE IMMEDIATE is incorrect.

The procedure will raise an exception because Dynamic SQL can drop tables but cannot drop views.

The procedure will raise an exception because one of the views is a complex view.

Incorrect. Refer to Section 9.

8. The following procedure adds a column of datatype DATE to the EMPLOYEES table. The name of the new column is passed to the procedure as a parameter.

```
CREATE OR REPLACE PROCEDURE addcol
    (p_col_name IN VARCHAR2) IS
    v_first_string VARCHAR2(100) := 'ALTER TABLE EMPLOYEES ADD (';
```

```
    v_second_string VARCHAR2(6) := ' DATE)';
BEGIN
    ... Line A
END;
```

Which of the following will work correctly when coded at line A? (Choose two.)
Mark for Review
(1) Points

(Choose all correct answers)

```
    v_first_string || p_col_name || v_second_string;
```

```
    EXECUTE IMMEDIATE v_first_string || p_col_name || v_second_string;
(*)
```

```
    EXECUTE IMMEDIATE 'v_first_string' || p_col_name || 'v_second_string';
```

```
    v_first_string := v_first_string || p_col_name;
EXECUTE IMMEDIATE v_first_string || v_second_string;
(*)
```

```
    EXECUTE v_first_string || p_col_name || v_second_string;
```

Correct

9. Which two of these declarations cannot be in the same package specification?

```
PROCEDURE myproc (p1 NUMBER, p2 VARCHAR2);
PROCEDURE myproc (p1 VARCHAR2, p2 NUMBER);
PROCEDURE myproc (p1 NUMBER, p2 CHAR);
PROCEDURE myproc (p1 NUMBER);
```

Mark for Review
(1) Points

1 and 2

1 and 3 (*)

2 and 3

3 and 4

1 and 4

Incorrect. Refer to Section 9.

10. A public function in a package is invoked from within a SQL statement. The function's code can include a COMMIT statement. True or False? Mark for Review
(1) Points

True

False (*)

Incorrect. Refer to Section 9.

Test: Mid Term Exam Semester 2

Review your answers, feedback, and question scores below. An asterisk (*) indicates a correct answer.

Section 9

11. Examine the following package code:

```
CREATE OR REPLACE PACKAGE ol_pack IS
    PROCEDURE subprog (p1 IN VARCHAR2, p2 IN NUMBER);
    PROCEDURE subprog (param1 IN CHAR, param2 IN NUMBER);
    FUNCTION subprog (param1 IN VARCHAR2, param2 IN NUMBER) RETURN DATE;
END ol_pack;
```

Which of the following calls will be successful? (Choose two.)
Mark for Review
(1) Points

(Choose all correct answers)

```
ol_pack.subprog('Jane',30);

ol_pack.subprog(param1=>'Jane',param2=>30); (*)

v_number := ol_pack.subprog(p1=>'Jane');

v_date := ol_pack.subprog('Jane',30); (*)
```

Incorrect. Refer to Section 9.

12. Package TAXPACK declares a global variable G_TAXRATE NUMBER(2,2). The value of the tax rate is stored in table TAXTAB in the database. You want to read this value automatically into G_TAXRATE each time a user session makes its first call to TAXPACK. How would you do this? Mark for Review
(1) Points

Declare the global variable as:

```
g_taxrate NUMBER(2,2) := SELECT tax_rate FROM taxtab;
```

Create a database trigger that includes the following code:

```
SELECT tax_rate INTO taxpack.g_taxrate FROM taxtab;
```

Add a private function to the package body of TAXPACK, and invoke the function from the user session.

Add a package initialization block to the package body of TAXPACK. (*)

Incorrect. Refer to Section 9.

13. Package MULTIPACK declares the following global variable:
g_myvar NUMBER;
User DICK executes the following:
multipack.g_myvar := 45;

User HAZEL now connects to the database. Both users immediately execute:

```
BEGIN
    DBMS_OUTPUT.PUT_LINE(multipack.g_myvar);
END;
```

What values will Dick and Hazel see?
Mark for Review
(1) Points

Dick: 45, Hazel: 45

Dick: 45, Hazel: 0

Dick: 45, Hazel: null (*)

Dick: 0, Hazel: 0

Both queries will fail because the syntax of DBMS_OUTPUT.PUT_LINE is incorrect

Incorrect. Refer to Section 9.

14. When a user session changes the value of a package variable, the new value can immediately be seen by other sessions. True or False? Mark for Review
(1) Points

True

False (*)

Incorrect. Refer to Section 9.

15. The DBMS_OUTPUT.PUT procedure places text in a buffer but does not display the contents of the buffer. True or False? Mark for Review
(1) Points

True (*)

False

Correct

16. An Oracle directory called FILESDIR has been created by executing:
CREATE OR REPLACE DIRECTORY filesdir AS 'C:\NEWFILES';
Which of the following will create a new text file called C:\NEWFILES\EMP_REPORT.TXT ? Mark for Review

(1) Points

```
UTL_FILE.CREATE('FILESDIR','EMP_REPORT.TXT');

UTL_FILE.FOPEN('C:\NEWFILES\EMP_REPORT.TXT','w');

UTL_FILE.FOPEN('FILESDIR','EMP_REPORT.TXT','w'); (*)

UTL_FILE.OPEN('FILESDIR','EMP_REPORT.TXT','c');
```

Incorrect. Refer to Section 9.

17. Which of the following best describes the purpose of the UTL_FILE package? Mark for Review

(1) Points

It is used to load binary files such as employees' photos into the database.

It is used to read and write text files stored outside the database. (*)

It is used to find out how much free space is left on an operating system disk.

It is used to query CHAR and VARCHAR2 columns in tables.

Incorrect. Refer to Section 9.

18. Why is it better to use DBMS_OUTPUT only in anonymous blocks, not inside stored subprograms such as procedures? Mark for Review

(1) Points

Because DBMS_OUTPUT cannot be used inside procedures

Because anonymous blocks display messages while the block is executing, while procedures do not display anything until their execution has finished

Because DBMS_OUTPUT should be used only for testing and debugging PL/SQL code (*)

Because DBMS_OUTPUT can raise a NO_DATA_FOUND exception if used inside a packaged procedure

Correct

19. We need to declare a package variable named MYVAR, which can be referenced by any subprogram in the package but can NOT be referenced from outside the package. In the following code, where should MYVAR be declared?

```
CREATE OR REPLACE PACKAGE varpack IS
    -- Point A
...
END varpack;
CREATE OR REPLACE PACKAGE BODY varpack IS
    -- Point B
PROCEDURE varproc IS
    -- Point C
    BEGIN
       ...
    END varproc;
```

```
PROCEDURE ...
...
    -- Point D
END varpack;
```

Mark for Review
(1) Points

Point A

Point B (*)

Point C

Point D

Point B or Point C, they will both work

Incorrect. Refer to Section 9.

20. Package NEWPACK contains several procedures and functions, including private function PRIVFUNC. From where can PRIVFUNC be invoked? (Choose two.) Mark for Review
(1) Points

(Choose all correct answers)

From an anonymous block

From any procedure in NEWPACK (*)

From any private function in another package

From any function in NEWPACK (*)

From any public procedure in another package

Correct

Test: Mid Term Exam Semester 2

Review your answers, feedback, and question scores below. An asterisk (*) indicates a correct answer.

Section 9

21. Your schema contains four packages, each having a specification and a body. You have also been granted privileges to access three packages (and their bodies) in other users' schemas. What will be displayed by the following query?

```
SELECT COUNT(*) FROM ALL_OBJECTS
    WHERE object_type LIKE 'PACK%'
    AND owner <> USER;
```

Mark for Review
(1) Points

14

7

3

6 (*)

0

Correct

22. Which of the following will display the detailed code of the subprograms in package DEPTPACK in your schema ? Mark for Review
(1) Points

```
    SELECT text FROM USER_SOURCE
WHERE name = 'DEPTPACK'
AND type = 'PACKAGE'
ORDER BY line;
```

```
    SELECT text FROM USER_SOURCE
WHERE name = 'DEPTPACK'
AND type = 'PACKAGE BODY'
ORDER BY line;
```
(*)

```
    SELECT text FROM USER_SOURCE
WHERE object_name = 'DEPTPACK'
AND object_type = 'PACKAGE BODY'
ORDER BY line;
```

```
    SELECT text FROM USER_SOURCE
WHERE name = 'DEPTPACK'
AND type = 'BODY'
ORDER BY line;
```

Correct

23. In a package, public components are declared in the specification but private components are not. True or False? Mark for Review
(1) Points

True (*)

False

Correct

24. Package OLDPACK is in your schema. What will happen when the following statement is executed?

```
DROP PACKAGE oldpack;
```

Mark for Review
(1) Points

The body will be dropped but the specification will be retained.

The specification will be dropped but the body will be retained.

Both the specification and the body will be dropped. (*)

The statement will fail because you must drop the body before you can drop the specification.

Incorrect. Refer to Section 9.

25. Examine the following package specification:

```
CREATE OR REPLACE PACKAGE taxpack IS
    CURSOR empcurs IS SELECT * FROM employees;
    PROCEDURE taxproc;
END mypack;
```

The package body of TAXPACK also includes a function called TAXFUNC. Which one of the following statements is NOT true?
Mark for Review
(1) Points

The procedure can be invoked by:

```
BEGIN
    taxpack.taxproc;
END;
```

The packaage will not compile because you cannot declare a cursor in the specification.
(*)

TAXPROC is a public procedure and TAXFUNC is a private function

TAXPROC can invoke TAXFUNC if TAXPROC is coded before TAXFUNC

TAXPROC can open the cursor

Incorrect. Refer to Section 9.

Section 10

26. Examine this code:

```
CREATE TRIGGER de_trigg
    -- Line A
BEGIN ...
```

Which of the following are NOT valid at Line A ? (Choose two.)
Mark for Review
(1) Points

(Choose all correct answers)

AFTER LOGOFF ON SCHEMA (*)

AFTER LOGON ON SCHEMA

BEFORE LOGOFF ON SCHEMA

BEFORE DISCONNECT ON SCHEMA (*)

AFTER SERVERERROR ON SCHEMA

Incorrect. Refer to Section 10.

27. You want to prevent any objects in your schema from being altered or dropped. You decide to create the following trigger:

```
CREATE TRIGGER stop_ad_trigg
    -- Line A
BEGIN
    RAISE_APPLICATION_ERROR(-20203,'Invalid Operation');
END;
```

What should you code at Line A ?
Mark for Review
(1) Points

AFTER ALTER OR DROP ON SCHEMA

INSTEAD OF ALTER OR DROP ON SCHEMA

BEFORE ALTER OR DROP ON SCHEMA (*)

BEFORE ALTER, DROP ON SCHEMA

AFTER ALTER, DROP ON SCHEMA

Incorrect. Refer to Section 10.

28. Examine this code:

```
CREATE TRIGGER new_trigg
AFTER CREATE ON reserved_word
BEGIN ...
```

Which of the following can be used in place of reserved_word? (Choose two.)
Mark for Review
(1) Points

(Choose all correct answers)

TABLE

SCHEMA (*)

USER

DATABASE (*)

TABLE employees

Correct

29. The database administrator wants to write a log record every time an Oracle Server error occurs in any user's session. The DBA creates the following trigger:

```
CREATE TRIGGER log_errs_trigg
    -- Line A
BEGIN
    INSERT INTO errlog_table VALUES (...);
END;
```

What should the DBA code at Line A ?

Mark for Review
(1) Points

AFTER ERROR ON DATABASE

AFTER SERVER ERROR ON DATABASE

AFTER SERVERERROR ON SCHEMA

AFTER SERVERERROR ON DATABASE (*)

AFTER ORACLE ERROR ON SCHEMA

Incorrect. Refer to Section 10.

30. Which kinds of trigger can cause a mutating table problem? (Choose two.)

Mark for Review
(1) Points

(Choose all correct answers)

BEFORE UPDATE row triggers (*)

DDL triggers

AFTER DELETE row triggers (*)

Database Event triggers

INSTEAD OF triggers

Incorrect. Refer to Section 10.

PlSQL feedback midterm semister 1 part2

Review your answers, feedback, and question scores below. An asterisk (*) indicates a correct answer.

Section 10

31. Examine the following code:

```
CREATE TRIGGER emp_trigg
AFTER UPDATE OF salary ON employees
FOR EACH ROW
DECLARE
    v_count NUMBER;
BEGIN
    -- Line A
END;
```

Which of the following statements is NOT allowed at Line A?
Mark for Review
(1) Points

SELECT count(*) INTO v_count FROM departments;

UPDATE employees SET job_id = 'IT_PROG' WHERE employee_id = :OLD.employee_id;

SELECT count(*) INTO v_count FROM employees; (*)

DBMS_OUTPUT.PUT_LINE('A salary was updated');

None. All of the above are allowed.

Incorrect. Refer to Section 10.

32. Which of the following statements could cause a DDL trigger to fire? Mark for Review
(1) Points

DROP TABLE employees;

ALTER TABLE departments ADD (budget NUMBER(8,2));

CREATE TABLE newemp AS SELECT * FROM employees;

TRUNCATE TABLE locations;

All of the above (*)

Incorrect. Refer to Section 10.

33. Which of the following are NOT stored inside the database? (Choose two.)
Mark for Review
(1) Points

(Choose all correct answers)

A PL/SQL package specification

A database trigger

An anonymous block (*)

An application trigger (*)

A sequence

Incorrect. Refer to Section 10.

34. A trigger can be a public subprogram within a PL/SQL package. True or False? Mark for Review
(1) Points

True

False (*)

Incorrect. Refer to Section 10.

35. A business rule states that an employee's salary cannot be greater than 99,999.99 or less than 0. The best way to enforce this rule is by using: Mark for Review
(1) Points

A datatype of NUMBER(7,2) for the SALARY column

A database trigger

A check constraint (*)

An application trigger

A view

Incorrect. Refer to Section 10.

36. A trigger can be created in the database or within an application. True or False? Mark for Review
(1) Points

True (*)

False

Correct

37. The following objects have been created in a user's schema:
- a function FUNC1
- A package PACK1 which contains a public procedure PACKPROC and a private function PACKFUNC
- a trigger TRIGG1.
The procedure and functions each accept a single IN parameter of type NUMBER, and the functions return BOOLEANs. Which of the following calls to these objects (from an anonymous block) are correct? (Choose two.) Mark for Review
(1) Points

(Choose all correct answers)

pack1.packproc(25); (*)

SELECT func1(100) FROM dual;

trigg1;

IF pack1.packfunc(40) THEN ...

IF func1(75) THEN ... (*)

Incorrect. Refer to Section 10.

38. You can use a trigger to prevent rows from being deleted from the EMPLOYEES table on Mondays. True or False? Mark for Review
(1) Points

True (*)

False

Correct

39. Which of the following best describes a database trigger? Mark for Review
(1) Points

A subprogram that checks whether a user has typed the correct password to log on to the database.

A PL/SQL subprogram that executes automatically whenever an associated database event occurs. (*)

A PL/SQL subprogram that always returns exactly one value.

A subprogram that is invoked explicitly by the calling application.

A PL/SQL subprogram that inserts rows into a logging table.

Incorrect. Refer to Section 10.

40. In the following code:
CREATE TRIGGER mytrigg
INSTEAD OF INSERT OR UPDATE ON my_object_name
FOR EACH ROW
BEGIN ...
my_object_name can be the name of a table. True or False?
Mark for Review
(1) Points

True

False (*)

Incorrect. Refer to Section 10.

Test: Mid Term Exam Semester 2

Review your answers, feedback, and question scores below. An asterisk (*) indicates a correct answer.

Section 10

41. Examine the following trigger. It should raise an application error if a user tries to update an employee's last name. It should allow updates to all other columns of the EMPLOYEES table. What should be coded at line A?

```
CREATE TRIGGER stop_ln_trigg
BEFORE UPDATE ON employees
BEGIN
       -- Line A
    RAISE_APPLICATION_ERROR(-20201,'Updating last name not allowed');
    END IF;
END;
```

Mark for Review
(1) Points

IF UPDATING LAST_NAME THEN

IF UPDATING('LAST_NAME') THEN (*)

IF UPDATE('LAST_NAME') THEN

IF UPDATING THEN

Correct

42. Examine the following code. To create a row trigger, what code should be included at Line A?

```
CREATE TRIGGER dept_trigg
AFTER UPDATE OR DELETE ON departments
    -- Line A
BEGIN ...
```

Mark for Review
(1) Points

AFTER EACH ROW

FOR EVERY ROW

FOR EACH ROW (*)

ON EACH ROW

ON EVERY ROW

Incorrect. Refer to Section 10.

43. With which kind of trigger can the :OLD and :NEW qualifiers be used? Mark for Review
(1) Points

DDL triggers

Database Event triggers

Statement triggers

Row triggers (*)

AFTER triggers

Incorrect. Refer to Section 10.

44. A DML statement trigger fires only once for each triggering DML statement, while a row trigger fires once for each row processed by the triggering statement. True or False? Mark for Review
(1) Points

True (*)

False

Correct

45. You need to create a trigger that will fire whenever an employee's salary or job_id is updated, but not when any other column of the EMPLOYEES table is updated. Which of the following is the correct syntax to do this? Mark for Review
(1) Points

CREATE TRIGGER emp_upd_trigg
AFTER UPDATE ON employees (salary, job_id)
BEGIN ...

CREATE TRIGGER emp_upd_trigg
AFTER UPDATE OF salary, job_id ON employees
BEGIN ...
(*)

CREATE TRIGGER emp_upd_trigg
AFTER UPDATE OF (salary, job_id) ON employees
BEGIN ...

CREATE TRIGGER emp_upd_trigg
AFTER UPDATE OF salary OR job_id ON employees
BEGIN ...

Incorrect. Refer to Section 10.

46. The following code will successfully create emp_trigg: True or False?

```
CREATE OR REPLACE TRIGGER emp_trigg
BEFORE DELETE OF salary ON employees
BEGIN
    RAISE_APPLICATION_ERROR(-20202,'Deleting salary is not allowed');
END;
```

Mark for Review
(1) Points

True

False (*)

Incorrect. Refer to Section 10.

47. What is wrong with the following code?

```
CREATE TRIGGER dept_trigg
BEFORE UPDATE OF department_name ON departments
BEGIN
    DBMS_OUTPUT.PUT_LINE(:NEW.department_name);
END;
```

Mark for Review
(1) Points

You cannot use :NEW in a BEFORE trigger, only in an AFTER trigger.

You cannot use :NEW or :OLD in a statement trigger. (*)

You cannot use DBMS_OUTPUT.PUT_LINE inside a trigger.

The second line should be:
BEFORE UPDATE ON departments.department_name

Incorrect. Refer to Section 10.

48. You need to disable all triggers that are associated with DML statements on the DEPARTMENTS table. Which of the following commands should you use? Mark for Review
(1) Points

ALTER TABLE departments DISABLE ALL TRIGGERS; (*)

ALTER TRIGGER DISABLE ALL ON departments;

ALTER TABLE departments DISABLE TRIGGERS;

DISABLE ALL TRIGGERS ON departments;

ALTER TABLE departments DROP ALL TRIGGERS;

Incorrect. Refer to Section 10.

49. User AYSEGUL successfully creates the following trigger:

```
CREATE TRIGGER loc_trigg
BEFORE UPDATE ON aysegul.locations
BEGIN ....
```

AYSEGUL now tries to drop the LOCATIONS table. What happens?
Mark for Review
(1) Points

An error message is displayed because you cannot drop a table that is associated with a trigger.

The table is dropped and the trigger is disabled.

The trigger is dropped but the table is not dropped.

Both the table and the trigger are dropped. (*)

None of the above.

Incorrect. Refer to Section 10.

50. Which dictionary view shows the detailed code of a trigger body? Mark for Review
(1) Points

USER_SOURCE

USER_TRIGGERS (*)

USER_OBJECTS

USER_DML_TRIGGERS

USER_SUBPROGRAMS

Incorrect. Refer to Section 10.

Test: Mid Term Exam Semester 1 - Part II

Review your answers, feedback, and question scores below. An asterisk (*) indicates a correct answer.

PlSQL feedback midterm semister 1 part2

The Mid Term Exam for Semester 1 is presented to you as two exams. This is Part II of the Mid Term Exam for Semester 1.

Section 4

1. What will be the value of variable c after the following code is executed?

```
DECLARE
    a BOOLEAN := TRUE;
    b BOOLEAN := NULL;
    c NUMBER;
BEGIN
    IF a AND b THEN c := 2;
       ELSIF a OR b THEN c := 0;
       ELSE c := 1;
    END IF;
END;
```

Mark for Review
(1) Points

1

Null

0 (*)

2

Correct

2. Which of the following is NOT a characteristic of a CASE statement? Mark for Review
(1) Points

It ends with END CASE;

It can be a complete PL/SQL block

It returns a value (*)

It evaluates a condition and performs an action

Incorrect. Refer to Section 4.

3. What will be the value of variable c after the following code is executed?

```
DECLARE
    a BOOLEAN := TRUE;
    b BOOLEAN := FALSE;
    c NUMBER;
BEGIN
    c :=
        CASE
            WHEN a AND b THEN 10
            WHEN NOT a THEN 20
            WHEN a OR b THEN 30
            ELSE 40
        END;
END;
```

Mark for Review
(1) Points

30 (*)

20

40

10

Incorrect. Refer to Section 4.

4. You want to assign a value to v_result which depends on the value of v_grade: if v_grade = 'A' set v_result to 'Very Good' and so on.

```
DECLARE
    v_grade CHAR(1);
    v_result VARCHAR2(10);
BEGIN
    v_result :=
       CASE v_grade
```

The next line should be

Mark for Review
(1) Points

WHEN v_grade = 'A' THEN 'Very Good'

WHEN 'A' THEN 'Very Good';

WHEN 'A' THEN v_result := 'Very Good';

WHEN 'A' THEN 'Very Good' (*)

Incorrect. Refer to Section 4.

5. What value will v_answer contain after the following code is executed?

```
DECLARE
    v_age NUMBER:= 18;
    v_answer VARCHAR2(10);
BEGIN
    v_answer :=
       CASE
          WHEN v_age < 25 THEN 'Young'
          WHEN v_age = 18 THEN 'Exactly 18'
          ELSE 'Older'
       END CASE;
END;
```

Mark for Review
(1) Points

Exactly 18

Young (*)

Null

Older

Correct

6. What kind of statement is best suited for displaying the multiplication table for "sixes": 6x1=6, 6x2=12 ... 6x12=72? Mark for Review
(1) Points

CASE expression

IF statement

CASE statement

LOOP statement (*)

Incorrect. Refer to Section 4.

7. Examine the following block:

```
DECLARE
    v_counter PLS_INTEGER := 1;
BEGIN
    LOOP
       DBMS_OUTPUT.PUT_LINE(v_counter);
       v_counter := v_counter + 1;
    EXIT WHEN v_counter = 5;
    END LOOP;
END;
```

What is the last value of V_COUNTER that is displayed?
Mark for Review
(1) Points

5

6

4 (*)

This is an infinite loop; the loop will never finish.

Incorrect. Refer to Section 4.

8. A PL/SQL block contains the following code:

```
v_counter := 1;
LOOP
EXIT WHEN v_counter = 5;
    v_counter := v_counter + 1;
END LOOP;
```

What is the value of V_COUNTER after the loop is finished?
Mark for Review
(1) Points

5 (*)

6

1

This is an infinite loop; the loop will never finish.

Incorrect. Refer to Section 4.

9. Which kind of loop is this?

```
v_count := 1;
LOOP
    v_count := v_count + 1;
    EXIT WHEN i > 20;
END LOOP;
```

Mark for Review
(1) Points

FOR loop

IF-THEN loop

Basic loop (*)

WHILE loop

CASE loop

Incorrect. Refer to Section 4.

10. The EXIT statement can be located anywhere inside a basic loop. True or False? Mark for Review
(1) Points

True (*)

False

Correct

Test: Mid Term Exam Semester 1 - Part II

Review your answers, feedback, and question scores below. An asterisk (*) indicates a correct answer.

The Mid Term Exam for Semester 1 is presented to you as two exams. This is Part II of the Mid Term Exam for Semester 1.

Section 4

11. What will happen when the following code is executed?

```
BEGIN
FOR i in 1 ..3 LOOP
    DBMS_OUTPUT.PUT_LINE (i);
    i := i + 1;
END LOOP;
END;
```

Mark for Review
(1) Points

It will display 1, 2, 3.

It will display 2, 3, 4.

It will result in an error because you cannot modify the counter in a FOR loop. (*)

It will result in an error because the counter was not explicitly declared.

Incorrect. Refer to Section 4.

12. When using a counter to control a FOR loop, which of the following is true ? Mark for Review
(1) Points

You must have exactly one counter but it is implicitly declared. (*)

You must have exactly one counter and you must explicitly declare it.

You can have multiple counters, but you need at least one.

You don't need a counter; you can test for anything (for example, whether a BOOLEAN is TRUE or FALSE).

Incorrect. Refer to Section 4.

13. In a WHILE loop, the statements inside the loop must execute at least once. True or False? Mark for Review
(1) Points

True

False (*)

Incorrect. Refer to Section 4.

14. Which statement best describes when a WHILE loop shouild be used? Mark for Review
(1) Points

When the number of iterations is known

When repeating a sequence of statements until the controlling condition is no longer true (*)

When assigning a value to a Boolean variable

When testing whether a variable is null

Correct

15. Which of the following blocks produces the same output as this block?

```
BEGIN
    FOR i in 1 .. 3 LOOP
       DBMS_OUTPUT.PUT_LINE(i);
    END LOOP;
END;
```

Mark for Review
(1) Points

```
    DECLARE
    i PLS_INTEGER := 0;
BEGIN
    WHILE i<3 LOOP
       DBMS_OUTPUT.PUT_LINE(i);
       i := i + 1;
    END LOOP;
END;
```

```
    DECLARE
    i PLS_INTEGER := 0;
BEGIN
    WHILE i<3 LOOP
       i := i + 1;
       DBMS_OUTPUT.PUT_LINE(i);
    END LOOP;
END;
(*)
```

```
    DECLARE
    i PLS_INTEGER := 0;
BEGIN
    WHILE i<3 LOOP
       DBMS_OUTPUT.PUT_LINE(i);
    END LOOP;
    i := i+ 1;
END;
```

Incorrect. Refer to Section 4.

16. In a FOR loop, an implicitly declared counter automatically increases or decreases with each iteration. True or False? Mark for Review
(1) Points

True (*)

False

Correct

17. Examine the following code:

```
DECLARE
    a BOOLEAN := TRUE;
    b BOOLEAN := FALSE;
    c BOOLEAN := TRUE;
    d BOOLEAN := FALSE;
    game char(4) := 'lost';
BEGIN
    IF ((a AND b) AND (c OR d))
    THEN game := 'won';
    END IF;
```

What is the value of GAME at the end of this block?
Mark for Review
(1) Points

NULL

won'

lost' (*)

False

Incorrect. Refer to Section 4.

18. Examine the following code:

```
DECLARE
    v_salary NUMBER(6);
    v_constant NUMBER(6) := 15000;
    v_result VARCHAR(6); := 'MIDDLE';
BEGIN
    IF v_salary != v_constant THEN
        v_result := 'HIGH';
    ELSE
        v_result := 'LOW';
    END IF;
END;
```

What is the final value of v_result?
Mark for Review
(1) Points

HIGH

LOW (*)

MIDDLE

Null

Incorrect. Refer to Section 4.

19. What is the correct name for CASE, LOOP, WHILE, and IF-THEN-ELSE structures ?
Mark for Review
(1) Points

Control structures (*)

Array structures

Memory structures

Cursor structures

Incorrect. Refer to Section 4.

20. What is the correct form of a simple IF statement? Mark for Review (1) Points

```
IF condition THEN statement;
```

```
IF condition THEN statement;
END IF; (*)
```

```
IF condition;
THEN statement;
END IF;
```

```
IF condition
THEN statement
ENDIF;
```

Correct

Test: Mid Term Exam Semester 1 - Part II

Review your answers, feedback, and question scores below. An asterisk (*) indicates a correct answer.

The Mid Term Exam for Semester 1 is presented to you as two exams. This is Part II of the Mid Term Exam for Semester 1.

Section 4

21. Examine the following code:

```
DECLARE
    a VARCHAR2(6) := NULL;
    b VARCHAR2(6) := NULL;
BEGIN
    IF a = b THEN
       DBMS_OUTPUT.PUT_LINE('EQUAL');
    ELSIF a != b THEN
       DBMS_OUTPUT.PUT_LINE('UNEQUAL');
```

```
    ELSE
       DBMS_OUTPUT.PUT_LINE('OTHER');
    END IF;
END;
```

Which word will be displayed?
Mark for Review
(1) Points

UNEQUAL

EQUAL

Nothing will be displayed

OTHER (*)

Incorrect. Refer to Section 4.

22. Examine the following code:

```
BEGIN
FOR i IN 1..5 LOOP
FOR j IN 1..8 LOOP
EXIT WHEN j = 7;
DBMS_OUTPUT.PUT_LINE(i || j);
END LOOP;
END LOOP;
END;
```

How many lines of output will be displayed when this code is executed? Mark for Review
(1) Points

35

6

30 (*)

40

Incorrect. Refer to Section 4.

23. In the following code fragment, you want to exit from the outer loop at Line A if v_number = 6. Which statement would you write on Line A?

```
<<big_loop>>
WHILE condition_1 LOOP
    <<small_loop>>
    FOR i IN 1..10 LOOP
       DBMS_OUTPUT.PUT_LINE(i);
       -- Line A
    END LOOP;
END LOOP;
```

Mark for Review
(1) Points

IF v_number = 6 THEN EXIT;

EXIT outer_loop WHEN v_number = 6;

```
    EXIT big_loop WHEN v_number = 6; (*)

    EXIT small_loop WHEN v_number = 6;
```

Incorrect. Refer to Section 4.

24. Examine the following code:

```
DECLARE
    v_outer_count NUMBER := 1;
    v_inner_count NUMBER := 1;
BEGIN
    LOOP
       LOOP
          v_inner_count := v_inner_count + 1;
          EXIT WHEN v_inner_count > 5; -- Line A
       END LOOP;
       v_outer_count := v_outer_count + 1;
       EXIT WHEN v_outer_count > 3;
    END LOOP;
END;
```

What happens at Line A when the value of V_INNER_COUNT equals 6?
Mark for Review
(1) Points

Both loops are exited and the block's execution is terminated.

The inner loop is exited but the outer loop continues execution. (*)

The outer loop is exited but the inner loop continues execution.

An error condition is returned.

Correct

25. When coding two nested loops, both loops must be of the same type. For example, you cannot code a FOR loop inside a WHILE loop. True or False? Mark for Review
(1) Points

True

False (*)

Incorrect. Refer to Section 4.

Section 5

26. You want to declare a cursor which locks each row fetched by the cursor. Examine the following code:

```
DECLARE
CURSOR emp_curs IS
SELECT * FROM employees
FOR -- Point A
```

Which of the following can NOT be coded at Point A? Mark for Review
(1) Points

UPDATE;

UPDATE OF salary;

UPDATE OF employees; (*)

UPDATE NOWAIT;

Incorrect. Refer to Section 5.

27. A cursor is declared as:
CURSOR c IS SELECT * FROM departments FOR UPDATE;
After opening the cursor and fetching some rows, you want to delete the most recently fetched row. Which of the following will do this successfully? Mark for Review
(1) Points

DELETE FROM c WHERE CURRENT OF c;

DELETE FROM departments WHERE CURRENT OF c; (*)

DELETE FROM c WHERE CURRENT OF departments;

DELETE FROM departments WHERE c%ROWCOUNT = 1;

None of the above.

Incorrect. Refer to Section 5.

28. Examine the following code:

```
DECLARE
CURSOR c IS SELECT * FROM employees FOR UPDATE;
c_rec c%ROWTYPE;
BEGIN
OPEN c;
FOR i IN 1..20 LOOP
FETCH c INTO c_rec;
IF i = 6 THEN
UPDATE employees SET first_name = 'Joe'
WHERE CURRENT OF c;
END IF;
END LOOP;
CLOSE c;
END;
```

Which employee row or rows will be updated when this block is executed? Mark for Review
(1) Points

The first 6 fetched rows will be updated.

No rows will be updated because you locked the rows when the cursor was opened.

The 6th fetched row will be updated. (*)

The block will not compile because the cursor should have been declared .... FOR UPDATE WAIT 5;

None of the above.

Incorrect. Refer to Section 5.

29. There are 12 distinct JOB_IDs in the EMPLOYEES table. You need to write some PL.SQL code to fetch and display all the employees with a specific JOB_ID. The chosen JOB_ID can be different each time the code is executed.
What is the best way to do this? Mark for Review
(1) Points

Write 12 separate PL/SQL blocks, each declaring a cursor with a different JOB_ID in the WHERE clause.

Write a single PL/SQL block which declares 12 cursors, one for each distinct value of JOB_ID.

Write a single PL/SQL block which declares one cursor using a parameter for the JOB_ID. (*)

Write a single PL/SQL block which uses a cursor to fetch all the employee rows, with an IF statement to decide which of the fetched rows to display.

Correct

30. A cursor has been declared as:
CURSOR c_curs (p_param VARCHAR2) IS
SELECT * FROM mytable
WHERE mycolumn = p_param.
Which of the following will open the cursor successfully? Mark for Review
(1) Points

OPEN c_curs(p_param = 'ABC');

OPEN c_curs('ABC'); (*)

OPEN c_curs USING ('ABC');

p_param := 'ABC';
OPEN c_curs(p_param);

Incorrect. Refer to Section 5.

Test: Mid Term Exam Semester 1 - Part II

Review your answers, feedback, and question scores below. An asterisk (*) indicates a correct answer.

PlSQL feedback midterm semister 1 part2

The Mid Term Exam for Semester 1 is presented to you as two exams. This is Part II of the Mid Term Exam for Semester 1.

Section 5

31. What is wrong with the following code?

```
BEGIN
    FOR emp_rec IN
       (SELECT * FROM employees WHERE ROWNUM < 10
          FOR UPDATE NOWAIT) LOOP
        DBMS_OUTPUT.PUT_LINE(emp_rec%ROWCOUNT || emp_rec.last_name):
    END LOOP;
END;
```

Mark for Review
(1) Points

You cannot use FOR UPDATE NOWAIT with a cursor FOR loop using a subquery.

You cannot reference %ROWCOUNT with a cursor FOR loop using a subquery. (*)

The field EMP_REC.LAST_NAME does not exist.

You cannot use ROWNUM with a cursor FOR loop.

The cursor has not been opened.

Incorrect. Refer to Section 5

32. The following code fragment shows a cursor FOR loop:

```
FOR emp_record IN emp_cursor LOOP ......
```

Which of the following do NOT need to be coded explicitly? (Choose three.)
Mark for Review
(1) Points

(Choose all correct answers)

OPEN emp_cursor; (*)

DECLARE CURSOR emp_cursor IS ...

emp_record emp_cursor%ROWTYPE; (*)

FETCH emp_cursor INTO emp_record; (*)

END LOOP;

Incorrect. Refer to Section 5

33. There are no employees in department_id 75.
Which of the following is NOT a valid cursor FOR loop with a subquery? Mark for Review
(1) Points

FOR emp_rec IN
(SELECT last_name, salary FROM employees) LOOP ...

```
FOR emp_rec IN
(SELECT * FROM employees) LOOP ...


FOR emp_rec IN
(SELECT last_name, salary FROM employees ORDER BY last_name) LOOP ...


FOR emp_rec IN
(SELECT * FROM employees WHERE department_id = 75) LOOP ...
```

None of the above. They are all valid. (*)

Incorrect. Refer to Section 5

34. When using multiple nested cursors, what kinds of loops can you use? Mark for Review
(1) Points

Cursor FOR loops only.

Basic loops only.

WHILE loops only.

None of the above.

All of the above. (*)

Incorrect. Refer to Section 5.

35. What is wrong with the following code?

```
DECLARE
CURSOR emp_curs(p_dept_id NUMBER) IS
SELECT * FROM employees WHERE department_id = p_dept_id;
BEGIN
FOR dept_rec IN (SELECT * FROM departments) LOOP
DBMS_OUTPUT.PUT_LINE(dept_rec.department_name);
FOR emp_rec IN emp_curs(dept_rec.department_id) LOOP
DBMS_OUTPUT.PUT_LINE(emp_rec.last_name);
END LOOP;
END LOOP;
END;
```

Mark for Review
(1) Points

The DEPARTMENTS cursor must be declared with a parameter.

You cannot use a cursor with a subquery in nested loops.

You cannot use two different kinds of loop in a single PL/SQL block.

EMP_CURS should not be DECLAREd explicitly; it should be coded as a subquery in a cursor FOR loop.

Nothing is wrong. The block will execute successfully and display all

departments and the employees in those departments. (*)

Incorrect. Refer to Section 5.

36. How many explicit cursors can be declared and used in a single PL/SQL block? Mark for Review
(1) Points

One or two.

Only one.

As many as needed. (*)

Up to eight cursors.

None of the above.

Incorrect. Refer to Section 5.

37. Assume that you have declared a cursor called C_EMP. Which of the following statements about C_EMP is correct? (Choose two.) Mark for Review
(1) Points

(Choose all correct answers)

You can use c_emp%NOTFOUND to exit a loop. (*)

You can fetch rows when c_emp%ISOPEN evaluates to FALSE.

You can use c_emp%ROWCOUNT to return the number of rows returned by the cursor so far. (*)

You can use c_emp%FOUND after the cursor is closed.

Correct

38. Which of the following cursor attributes is set to the total number of rows returned so far? Mark for Review
(1) Points

%ISOPEN

%NOTFOUND

%FOUND

%ROWCOUNT (*)

Correct

39. The employees table contains 11 columns. The following block declares a cursor and a record based on the cursor:

```
DECLARE
    CURSOR emp_curs IS
```

```
    SELECT * FROM employees;
  v_emp_rec emp_curs%ROWTYPE;
```

A twelfth column is now added to the employees table. Which of the following statements is true?
Mark for Review
(1) Points

The declaration of emp_rec must be changed to add an extra field.

The block will still work correctly without any changes to the PL/SQL code. (*)

The block will fail and an INVALID_CURSOR exception will be raised.

An extra scalar variable must be declared to correspond to the twelfth table column.

Incorrect. Refer to Section 5.

40. The DEPARTMENTS table contains four columns. Examine the following code:

```
DECLARE
    CURSOR dept_curs IS
       SELECT * FROM departments;
    v_dept_rec dept_curs%ROWTYPE;
BEGIN
    OPEN dept_curs;
    FETCH dept_curs INTO v_dept_rec;
    ...
```

Which one of the following statements is true?
Mark for Review
(1) Points

v_dept_rec contains the first four rows of the departments table.

The FETCH will fail because the structure of v_dept_rec does not match the structure of the cursor.

v_dept_rec contains the first row of the departments table. (*)

The block will fail because the declaration of v_dept_rec is invalid.

Incorrect. Refer to Section 5.

Test: Mid Term Exam Semester 1 - Part II

Review your answers, feedback, and question scores below. An asterisk (*) indicates a correct answer.

PlSQL feedback midterm semister 1 part2

The Mid Term Exam for Semester 1 is presented to you as two exams. This is Part II of the Mid Term Exam for Semester 1.

Section 5

41. Which of the following cursor attributes evaluates to TRUE if the cursor is open? Mark for Review
(1) Points

%ISOPEN (*)

%NOTFOUND

%FOUND

%ROWCOUNT

Correct

42. Examine the following code fragment:

```
DECLARE
    CURSOR emp_curs IS
       SELECT first_name, last_name FROM employees;
    v_emp_rec emp_curs%ROWTYPE;
BEGIN
    ...
    FETCH emp_curs INTO v_emp_rec;
    DBMS_OUTPUT.PUT_LINE(... Point A ...);
   ...
```

To display the fetched last name, what should you code at Point A?
Mark for Review
(1) Points

v_emp_rec.last_name (*)

v_emp_rec(last_name)

v_emp_rec

last_name

None of the above

Incorrect. Refer to Section 5.

43. An implicit cursor can be used for a multiple-row SELECT statement. True or False? Mark for Review
(1) Points

True

False (*)

Incorrect. Refer to Section 5.

44. The employees table contains 20 rows. What will happen when the following code is executed?

```
DECLARE
   CURSOR emp_curs IS
      SELECT job_id FROM employees;
    v_job_id employees.job_id%TYPE;
BEGIN
    OPEN emp_curs;
    LOOP
       FETCH emp_curs INTO v_job_id;
       DBMS_OUTPUT.PUT_LINE(v_job_id);
       EXIT WHEN emp_curs%NOTFOUND;
    END LOOP;
    CLOSE emp_curs;
END;
```

Mark for Review
(1) Points

20 job_ids will be displayed.

The block will fail and an error message will be displayed.

21 rows of output will be displayed; the first job_id will be displayed twice.

21 rows of output will be displayed; the last job_id will be displayed twice.
(*)

Incorrect. Refer to Section 5.

45. Which of these is NOT a valid cursor declaration? Mark for Review
(1) Points

```
CURSOR emp_curs IS
SELECT salary
FROM employees
ORDER BY salary DESC;
```

```
CURSOR emp_curs IS
SELECT salary
FROM employees
WHERE last_name LIKE 'S%';
```

```
CURSOR emp_dept_curs IS
SELECT e.salary, d.department_name
FROM employees e, departments d
WHERE e.department_id = d.department_id;
```

```
CURSOR emp_curs IS
SELECT salary INTO v_salary
FROM employees;
```
(*)

Incorrect. Refer to Section 5.

46. Examine the following code:

```
DECLARE
    CURSOR emp_curs IS
       SELECT last_name, salary
       FROM employees
       ORDER BY salary;
    v_last_name employees.last_name%TYPE;
    v_salary employees.salary%TYPE;
BEGIN
...
```

Which of the following statements successfully opens the cursor and fetches the first row of the active set?
Mark for Review
(1) Points

```
    OPEN emp_curs;
FETCH emp_curs INTO v_last_name, v_salary;
```
(*)

```
    OPEN emp_curs;
FETCH emp_curs INTO v_salary, v_last_name;
```

```
    OPEN emp_curs;
FETCH FIRST emp_curs INTO v_last_name, v_salary;
```

```
    OPEN emp_curs;
FETCH emp_curs;
```

Incorrect. Refer to Section 5.

47. Which of these statements about implicit cursors is NOT true? Mark for Review
(1) Points

They are declared automatically by Oracle for single-row SELECT statements.

They are declared automatically by Oracle for all DML statements.

They are declared by the PL/SQL programmer. (*)

They are opened and closed automatically by Oracle.

Incorrect. Refer to Section 5.

48. Which of these constructs can be used to fetch multiple rows from a cursor's active set? Mark for Review
(1) Points

A CASE statement

An IF .... ELSE statement

A basic loop which includes FETCH and EXIT WHEN statements (*)

A basic loop which includes OPEN, FETCH and CLOSE statements

Incorrect. Refer to Section 5.

49. Place the following statements in the correct sequence:

```
OPEN my_curs;
CLOSE my_curs;
CURSOR my_curs IS SELECT my_column FROM my_table;
FETCH my_curs INTO my_variable;
```

Mark for Review
(1) Points

C,D,A,B

C,A,D,B (*)

A,C,D,B

C,A,B,D

Correct

50. After a cursor has been closed, it can be opened again in the same PL/SQL block. True or False? Mark for Review
(1) Points

True (*)

False

Incorrect. Refer to Section 5.

Test: Mid Term Exam Semester 1 - Part I

Review your answers, feedback, and question scores below. An asterisk (*) indicates a correct answer.

The Mid Term Exam for Semester 1 is presented to you as two exams. This is Part I of the Mid Term Exam for Semester 1.

Section 1

1. Using Oracle Application Express, you can create Web applications that include

PlSQL feedback midterm semister 1 part2

PL/SQL. True or False? Mark for Review
(1) Points

True (*)

False

Correct

2. Which of the following statements about exception handling in PL/SQL is false? Mark for Review
(1) Points

You can prepare for database exceptions by creating exception handlers.

You can prepare for application exceptions by creating exception handlers.

Exception handling code tells your program what to do when an error is encountered.

Exception handling code can be grouped together in a PL/SQL block.

None of the above (*)

Correct

3. PL/SQL can be used not only with an Oracle database, but also with any kind of relational database. True or False? Mark for Review
(1) Points

True

False (*)

Correct

4. The P in PL/SQL stands for: Mark for Review
(1) Points

Processing

Procedural (*)

Primary

Proprietary

Correct

5. A program which specifies a list of operations to be performed sequentially to achieve the desired result can be called: Mark for Review
(1) Points

declarative

nondeclarative

procedural (*)

low level

Correct

6. Which of the following statements about PL/SQL and SQL is true? Mark for Review
(1) Points

PL/SQL and SQL are both ANSI-compliant.

PL/SQL and SQL can be used with many types of databases, including Oracle.

PL/SQL and SQL are both Oracle proprietary programming languages.

PL/SQL allows basic program logic and control flow to be combined with SQL statements. (*)

Correct

7. What kind of block is defined by the following PL/SQL code?

```
BEGIN
    DBMS_OUTPUT.PUT_LINE('My first quiz');
END;
```

Mark for Review
(1) Points

procedure

subroutine

function

anonymous (*)

Correct

8. Which PL/SQL block type must return a value? Mark for Review
(1) Points

Anonymous

Function (*)

Procedure

Correct

9. Which keywords must be included in every PL/SQL block? (Choose two.) Mark for Review
(1) Points

PlSQL feedback midterm semister 1 part2

(Choose all correct answers)

DECLARE

END; (*)

EXCEPTION

BEGIN (*)

DBMS_OUTPUT.PUT_LINE

Incorrect. Refer to Section 1.

10. Given below are the parts of a PL/SQL block:

1. END;
2. EXCEPTION
3. DECLARE
4. BEGIN

Arrange the parts in order.

Mark for Review

(1) Points

2,1,4,3

3,4,2,1 (*)

3,2,4,1

4,3,2,1

Correct

Test: Mid Term Exam Semester 1 - Part I

Review your answers, feedback, and question scores below. An asterisk (*) indicates a correct answer.

The Mid Term Exam for Semester 1 is presented to you as two exams. This is Part I of the Mid Term Exam for Semester 1.

Section 1

11. In which part of the PL/SQL block are declarations of variables defined?

Mark for Review

(1) Points

Executable

Exception

Declarative (*)

Definition

Correct

12. What is the purpose of using DBMS_OUTPUT.PUT_LINE in a PL/SQL block? Mark for Review
(1) Points

To perform conditional tests

To allow a set of statements to be executed repeatedly

To display results to check if our code is working correctly (*)

To store new rows in the database

Correct

13. Errors are handled in the Exception part of the PL/SQL block. True or False? Mark for Review
(1) Points

True (*)

False

Correct

Section 2

14. A variable must have a value if NOT NULL is specified. True or False? Mark for Review
(1) Points

True (*)

False

Correct

15. 1. Null
2. False
3. True
4. 0
Which of the above can be assigned to a Boolean variable?
Mark for Review
(1) Points

2 and 3

2, 3 and 4

1, 2 and 3 (*)

1, 2, 3 and 4

Correct

16. If you are using the %TYPE attribute, you can avoid hard coding the: Mark for Review
(1) Points

Data type (*)

Table name

Column name

Constraint

Correct

17. Which of the following are valid identifiers? (Choose two.) Mark for Review
(1) Points

(Choose all correct answers)

Full Name

students_street_address (*)

v_code (*)

#hours

completion_%

Correct

18. Which of the following are valid identifiers? (Choose two.) Mark for Review
(1) Points

(Choose all correct answers)

yesterday (*)

yesterday's date

number_of_students_in_the_class

v$testresult (*)

#students

Correct

19. Reserved words can be used as identifiers. True or False? Mark for Review
(1) Points

True

False (*)

Correct

20. When a variable is defined using the NOT NULL keywords, the variable must contain a value. True or False? Mark for Review
(1) Points

True (*)

False

Correct

Test: Mid Term Exam Semester 1 - Part I

Review your answers, feedback, and question scores below. An asterisk (*) indicates a correct answer.

The Mid Term Exam for Semester 1 is presented to you as two exams. This is Part I of the Mid Term Exam for Semester 1.

Section 2

21. Identify which of the following assignment statements are valid. (Choose three.) Mark for Review
(1) Points

(Choose all correct answers)

v_last_name := Chandra;

v_blackout_date := '31-DEC-2006'; (*)

v_population := 333444; (*)

v_music_type := 'ROCK'; (*)

Correct

22. Assignment statements can continue over several lines in PL/SQL. True or False? Mark for Review
(1) Points

True (*)

False

Correct

23. When a variable is defined using the CONSTANT keyword, the value of the variable cannot change. True or False? Mark for Review
(1) Points

True (*)

False

Correct

24. Variables can be used in the following ways in a PL/SQL block. (Choose two.) Mark for Review
(1) Points

(Choose all correct answers)

To store data values. (*)

To rename tables and columns.

To refer to a single data value several times. (*)

To comment code.

Correct

25. A collection is a composite data type. True or False? Mark for Review
(1) Points

True (*)

False

Correct

26. A movie is an example of which category of data type? Mark for Review
(1) Points

Scalar

Composite

Reference

LOB (*)

Correct

27. Which of these are PL/SQL data types? (Choose three.) Mark for Review
(1) Points

(Choose all correct answers)

Scalar (*)

Identifier

Delimiter

Composite (*)

LOB (*)

Correct

28. When an exception occurs within a PL/SQL block, the remaining statements in the executable section of the block are skipped. True or False? Mark for Review
(1) Points

True (*)

False

Correct

29. When nested blocks are used, which blocks can or must be labeled? Mark for Review
(1) Points

The inner block must be labeled, the outer block can be labeled.

Both blocks must be labeled

Nested blocks cannot be labeled

The outer block must be labeled if it is to be referred to in the inner block. (*)

Correct

30. Examine the following code. Line A causes an exception. What will be displayed when the block is executed?

```
DECLARE
    var_a NUMBER := 6;
    var_b DATE;
BEGIN
    var_a := var_a * 2;
```

```
    var_b := '28 December 2006'; -- Line A
    var_a := var_a * 2;
EXCEPTION
    WHEN OTHERS THEN
       DBMS_OUTPUT.PUT_LINE(var_a);
END;
```

Mark for Review
(1) Points

12 (*)

24

6

Nothing will be displayed

Correct

Test: Mid Term Exam Semester 1 - Part I

Review your answers, feedback, and question scores below. An asterisk (*) indicates a correct answer.

The Mid Term Exam for Semester 1 is presented to you as two exams. This is Part I of the Mid Term Exam for Semester 1.

Section 2

31. An exception occurs within the inner block of two nested blocks. The inner block does not have an EXCEPTION section. What always happens? Mark for Review
(1) Points

Both blocks fail and an error message is displayed by the calling environment

The exception is propagated to the outer block (*)

Oracle automatically tries to re-execute the inner block

The user's database session is automatically disconnected

Correct

32. What will be displayed when the following code is executed?

```
DECLARE
    x VARCHAR2(6) := 'Chang';
BEGIN
    DECLARE
```

```
   x VARCHAR2(12) := 'Susan';
BEGIN
   x := x || x;
END;
DBMS_OUTPUT.PUT_LINE(x);
END;
```

Mark for Review
(1) Points

Susan

Chang (*)

ChangChang

SusanChang

The code will fail with an error

Correct

33. The implicit data type conversion at Point A may not work correctly. Why not?

```
DECLARE
    v_mydate DATE;
BEGIN
    V_MYDATE := '29-Feb-04'; -- Point A
END;
```

Mark for Review
(1) Points

There are only 28 days in February

Oracle cannot implicitly convert a character string to a date, even if the string contains a valid date value

If the database language is not English, 'Feb' has no meaning. (*)

V_MYDATE has been entered in uppercase

Correct

34. What is wrong with this assignment statement?

```
myvar :=  'To be or not to be';

'That is the question';
```

Mark for Review
(1) Points

An assignment statement must be a single line of code

Nothing is wrong, the statement is fine

An assignment statement must have a single semicolon at the end (*)

"myvar" is not a valid name for a variable

Character literals should not be enclosed in quotes

Correct

35. What is the output when the following program is executed?

```
set serveroutput on
DECLARE
    a VARCHAR2(10) := '333';
    b VARCHAR2(10) := '444';
    c PLS_INTEGER;
    d VARCHAR2(10);
BEGIN
    c := TO_NUMBER(a) + TO_NUMBER(b);
    d := a || b;
    DBMS_OUTPUT.PUT_LINE(c);
    DBMS_OUTPUT.PUT_LINE(d);
END;
```

Mark for Review
(1) Points

Nothing. The code will result in an error.

c=777 and d=333444 (*)

c=777 and d=777

c=333444 and d=777

Correct

36. Single row character functions are valid SQL functions in PL/SQL. True or False? Mark for Review
(1) Points

True (*)

False

Correct

37. Which of the following are disadvantages of implicit data type conversions? (Choose two.) Mark for Review
(1) Points

(Choose all correct answers)

The code is harder to read and understand (*)

You cannot store alphabetic characters in a variable of data type NUMBER

If Oracle changes the conversion rules in the future, your code may not work any more (*)

Oracle cannot implicitly convert a number value to a character string

Incorrect. Refer to Section 2.

38. Examine the following code. What is the final value of V_MYVAR ?

```
DECLARE
    v_myvar NUMBER;
BEGIN
    v_myvar := 1 + 2 * 3;
    v_myvar := v_myvar * 2;
END;
```

Mark for Review
(1) Points

81

49

14 (*)

18

Correct

39. Examine the following code:

```
1 DECLARE
2 x NUMBER;
3 BEGIN
4 x:= '300';
5 END;
```

After line 4, what is the value of x?
Mark for Review
(1) Points

'300'

300 (*)

NULL

Correct

40. The DECODE function is available in PL/SQL procedural statements. True or False? Mark for Review
(1) Points

True

False (*)

Correct

Test: Mid Term Exam Semester 1 - Part I

Review your answers, feedback, and question scores below. An asterisk (*) indicates a correct answer.

The Mid Term Exam for Semester 1 is presented to you as two exams. This is Part I of the Mid Term Exam for Semester 1.

Section 3

41. How many DML statements can be included in a single transaction? Mark for Review
(1) Points

Only one

None. A transaction cannot include DML statements.

A maximum of four DML statements

As many as needed (*)

Correct

42. The following anonymous block of code is run:

```
BEGIN
    INSERT INTO countries (id, name)
    VALUES ('XA', 'Xanadu');
    SAVEPOINT XA;
    INSERT INTO countries (id, name)
    VALUES ('NV','Neverland');
    COMMIT;
    ROLLBACK TO XA;
END;
```

What happens when the block of code finishes?
Mark for Review
(1) Points

No data is inserted and no errors occur.

No data is inserted and an error occurs

Two rows are inserted and no errors occur.

Two rows are inserted and an error occurs. (*)

Correct

43. Assume there are 5 employees in Department 10. What happens when the following statement is executed?

```
UPDATE employees
SET salary=salary*1.1;
```

Mark for Review
(1) Points

All employees get a 10% salary increase. (*)

No rows are modified because you did not specify "WHERE department_id=10"

A TOO_MANY_ROWS exception is raised.

An error message is displayed because you must use the INTO clause to hold the new salary.

Correct

44. There are no employees in Department 77. What will happen when the following block is executed?

```
BEGIN
DELETE FROM employees
WHERE department_id=77;
DBMS_OUTPUT.PUT_LINE(SQL%ROWCOUNT)
END;
```

Mark for Review
(1) Points

A NO_DATA_FOUND exception is raised.

A NULL is displayed.

A zero (0) is displayed. (*)

An exception is raised because the block does not contain a COMMIT statement.

Incorrect. Refer to Section 3.

45. You declare an implicit cursor in the DECLARE section of a PL/SQL block. True or False? Mark for Review
(1) Points

True

False (*)

Correct

46. A variable is declared as:

```
DECLARE
    v_salary employees.salary%TYPE;
BEGIN
```

Which of the following is a correct use of the INTO clause?
Mark for Review
(1) Points

```
    SELECT salary
INTO v_salary
```

```
FROM employees
WHERE employee_id=100;
(*)


    SELECT v_salary
INTO salary
FROM employees
WHERE employee_id=100;


    SELECT salary
FROM employees
INTO v_salary;


    SELECT salary
FROM employees
WHERE employee_id=100
INTO v_salary;
```

Correct

47. Which one of these SQL statements can be directly included in a PL/SQL executable block? Mark for Review
(1) Points

```
    SELECT last_name FROM employees
WHERE employee_id=100;


    DESCRIBE employees;


    UPDATE employees
SET last_name='Smith';
(*)


    DROP TABLE employees;
```

Correct

48. Which rows will be deleted from the EMPLOYEES table when the following code is executed?

```
DECLARE
    salary employees.salary%TYPE := 12000;
BEGIN
    DELETE FROM employees
    WHERE salary > salary;
END;
```

Mark for Review
(1) Points

All rows whose SALARY column value is greater than 12000.

All rows in the table.

No rows. (*)

All rows whose SALARY column value is equal to 12000.

Correct

49. The following code will return the last name of the employee whose employee id is equal to 100: True or False?

```
DECLARE
    v_last_name employees.last_name%TYPE;
    employee_id employees.employee_id%TYPE := 100;
BEGIN
    SELECT last_name INTO v_last_name
    FROM employees
    WHERE employee_id = employee_id;
END;
```

Mark for Review
(1) Points

True

False (*)

Correct

50. Which one of these SQL statements can be directly included in a PL/SQL executable block? Mark for Review
(1) Points

```
    DELETE FROM employees
WHERE department_id=60;
```
(*)

```
    SELECT salary FROM employees
WHERE department_id=60;
```

```
    CREATE TABLE new_emps (last_name VARCHAR2(10), first_name VARCHAR2(10));
```

```
    DROP TABLE locations;
```

Correct

Score: 48 out of 50

Percentage Scored: 96 %

Mastery Score: 60 %

Test: Mid Term Exam Semester 1 - Part II

Review your answers, feedback, and question scores below. An asterisk (*) indicates a correct answer.

The Mid Term Exam for Semester 1 is presented to you as two exams. This is Part II of the Mid Term Exam for Semester 1.

Section 4

1. You want to display multiplication tables for numbers up to 12. The display should look like this:

```
1 x 1 = 1
1 x 2 = 2
.....
1 x 12 = 12
2 x 1 = 2
2 x 2 = 4
.....
2 x 12 = 24
3 x 1 = 3
.....
.....
12 x 12 = 144
```

Which of the following is an efficient way to do this in PL/SQL? Mark for Review
(1) Points

Use two nested FOR loops. (*)

Store all the numbers from 1 to 144 in a table, then fetch and display them using a cursor.

Create a function which accepts two numbers as IN parameters and returns their product. Invoke the function 144 times.

Write an anonymous block which contains 144 calls to DBMS_OUTPUT, each looking like: DBMS_OUTPUT.PUT_LINE('7 x 9 = 63');

Correct

2. Examine the following code:

```
DECLARE
    v_outer_count NUMBER := 1;
    v_inner_count NUMBER := 1;
BEGIN
    LOOP
       LOOP
          v_inner_count := v_inner_count + 1;
          EXIT WHEN v_inner_count > 5; -- Line A
       END LOOP;
       v_outer_count := v_outer_count + 1;
       EXIT WHEN v_outer_count > 3;
    END LOOP;
END;
```

What happens at Line A when the value of V_INNER_COUNT equals 6?
Mark for Review
(1) Points

Both loops are exited and the block's execution is terminated.

The inner loop is exited but the outer loop continues execution. (*)

The outer loop is exited but the inner loop continues execution.

An error condition is returned.

Correct

3. In the following code fragment, you want to exit from the outer loop at Line A if v_number = 6. Which statement would you write on Line A?

```
<<big_loop>>
WHILE condition_1 LOOP
    <<small_loop>>
    FOR i IN 1..10 LOOP
       DBMS_OUTPUT.PUT_LINE(i);
       -- Line A
    END LOOP;
END LOOP;
```

Mark for Review
(1) Points

IF v_number = 6 THEN EXIT;

EXIT outer_loop WHEN v_number = 6;

EXIT big_loop WHEN v_number = 6; (*)

EXIT small_loop WHEN v_number = 6;

Correct

4. Examine the following code:

```
BEGIN
FOR i IN 1..5 LOOP
FOR j IN 1..8 LOOP
EXIT WHEN j = 7;
DBMS_OUTPUT.PUT_LINE(i || j);
END LOOP;
END LOOP;
END;
```

How many lines of output will be displayed when this code is executed? Mark for Review
(1) Points

35

6

30 (*)

40

Correct

5. What will be the value of variable c after the following code is executed?

```
DECLARE
    a BOOLEAN := TRUE;
    b BOOLEAN := FALSE;
    c NUMBER;
BEGIN
    c :=
        CASE
            WHEN a AND b THEN 10
            WHEN NOT a THEN 20
            WHEN a OR b THEN 30
            ELSE 40
        END;
END;
```

Mark for Review
(1) Points

30 (*)

20

40

10

Correct

6. Which of the following is NOT a characteristic of a CASE statement? Mark for Review
(1) Points

It ends with END CASE;

It can be a complete PL/SQL block

It returns a value (*)

It evaluates a condition and performs an action

Correct

7. What will be the value of v_result after the following code is executed?

```
DECLARE
    v_grade CHAR(1) := NULL;
    v_result VARCHAR2(10);
BEGIN
    CASE v_grade
       WHEN 'A' THEN v_result := 'Very Good';
       WHEN 'F' THEN v_result := 'Poor';
       ELSE v_result := 'In Between';
    END;
END;
```

Mark for Review
(1) Points

Poor

In Between (*)

Null

Very Good

Correct

8. You want to display a message which depends on the value of v_grade: if v_grade = 'A' display 'Very Good', if v_grade = 'B' then display 'Good', and so on.

```
DECLARE
    v_grade CHAR(1);
BEGIN
    CASE v_grade
```

The next line should be
Mark for Review
(1) Points

WHEN 'A' THEN (*)

WHEN v_grade = 'A' THEN

WHEN 'A' THEN;

IF 'A' THEN

Correct

9. What will be the value of variable c after the following code is executed?

```
DECLARE
    a BOOLEAN := TRUE;
    b BOOLEAN := NULL;
    c NUMBER;
BEGIN
    IF a AND b THEN c := 2;
       ELSIF a OR b THEN c := 0;
       ELSE c := 1;
    END IF;
END;
```

Mark for Review
(1) Points

1

Null

0 (*)

2

Correct

10. Which kind of loop is this?

```
v_count := 1;
```

```
LOOP
    v_count := v_count + 1;
    EXIT WHEN i > 20;
END LOOP;
```

Mark for Review
(1) Points

FOR loop

IF-THEN loop

Basic loop (*)

WHILE loop

CASE loop

Correct

Test: Mid Term Exam Semester 1 - Part II

Review your answers, feedback, and question scores below. An asterisk (*) indicates a correct answer.

The Mid Term Exam for Semester 1 is presented to you as two exams. This is Part II of the Mid Term Exam for Semester 1.

Section 4

11. A PL/SQL block contains the following code:

```
v_counter := 1;
LOOP
    EXIT WHEN v_counter=5;
END LOOP;
v_counter := v_counter + 1;
```

What is the value of V_COUNTER after the loop is finished?
Mark for Review
(1) Points

5

6

1

This is an infinite loop; the loop will never finish. (*)

Incorrect. Refer to Section 4.

12. Which one of these is NOT a kind of loop? Mark for Review
(1) Points

ASCENDING loop (*)

FOR loop

Basic loop

WHILE loop

Correct

13. Examine the following block:

```
DECLARE
    v_counter PLS_INTEGER := 1;
BEGIN
    LOOP
       DBMS_OUTPUT.PUT_LINE(v_counter);
       v_counter := v_counter + 1;
    EXIT WHEN v_counter = 5;
    END LOOP;
END;
```

What is the last value of V_COUNTER that is displayed?
Mark for Review
(1) Points

5

6

4 (*)

This is an infinite loop; the loop will never finish.

Correct

14. The EXIT statement can be located anywhere inside a basic loop. True or False? Mark for Review
(1) Points

True (*)

False

Correct

15. In a FOR loop, an implicitly declared counter automatically increases or decreases with each iteration. True or False? Mark for Review
(1) Points

True (*)

False

Correct

16. Which of the following blocks produces the same output as this block?

```
BEGIN
    FOR i in 1 .. 3 LOOP
       DBMS_OUTPUT.PUT_LINE(i);
    END LOOP;
END;
```

Mark for Review
(1) Points

```
    DECLARE
    i PLS_INTEGER := 0;
BEGIN
    WHILE i<3 LOOP
       DBMS_OUTPUT.PUT_LINE(i);
       i := i + 1;
    END LOOP;
END;
```

```
    DECLARE
    i PLS_INTEGER := 0;
BEGIN
    WHILE i<3 LOOP
       i := i + 1;
       DBMS_OUTPUT.PUT_LINE(i);
    END LOOP;
END;
(*)
```

```
    DECLARE
    i PLS_INTEGER := 0;
BEGIN
    WHILE i<3 LOOP
       DBMS_OUTPUT.PUT_LINE(i);
    END LOOP;
    i := i+ 1;
END;
```

Correct

17. In a WHILE loop, the statements inside the loop must execute at least once. True or False? Mark for Review
(1) Points

True

False (*)

Correct

18. Which statement best describes when a WHILE loop shouild be used? Mark for

Review
(1) Points

When the number of iterations is known

When repeating a sequence of statements until the controlling condition is no longer true (*)

When assigning a value to a Boolean variable

When testing whether a variable is null

Correct

19. Examine the following code:

```
DECLARE
v_bool BOOLEAN := FALSE;
v_counter NUMBER(4) := 0;
BEGIN
... Line A
?
END;
```

Which of the following is NOT valid at line A?
Mark for Review
(1) Points

WHILE NOT v_boolean LOOP

WHILE v_boolean AND v_counter < 6 LOOP

WHILE v_counter > 8 LOOP

WHILE v_counter IN 1..5 LOOP (*)

Correct

20. Which statement best describes when a FOR loop should be used? Mark for Review
(1) Points

When the number of iterations is known (*)

When testing the value in a Boolean variable

When the controlling condition must be evaluated at the start of each iteration

Correct

Review your answers, feedback, and question scores below. An asterisk (*) indicates a correct answer.

PlSQL feedback midterm semister 1 part2

The Mid Term Exam for Semester 1 is presented to you as two exams. This is Part II of the Mid Term Exam for Semester 1.

Section 4

21. Examine the following code:

```
DECLARE
    a VARCHAR2(6) := NULL;
    b VARCHAR2(6) := NULL;
BEGIN
    IF a = b THEN
       DBMS_OUTPUT.PUT_LINE('EQUAL');
    ELSIF a != b THEN
       DBMS_OUTPUT.PUT_LINE('UNEQUAL');
    ELSE
       DBMS_OUTPUT.PUT_LINE('OTHER');
    END IF;
END;
```

Which word will be displayed?

Mark for Review

(1) Points

UNEQUAL

EQUAL

Nothing will be displayed

OTHER (*)

Correct

22. Examine the following code:

```
DECLARE
    v_salary NUMBER(6);
    v_constant NUMBER(6) := 15000;
    v_result VARCHAR(6); := 'MIDDLE';
BEGIN
    IF v_salary != v_constant THEN
        v_result := 'HIGH';
    ELSE
        v_result := 'LOW';
    END IF;
END;
```

What is the final value of v_result?

Mark for Review

(1) Points

HIGH

LOW (*)

MIDDLE

Null

Correct

23. How many ELSIF statements are you allowed to have in a compound IF statement? Mark for Review
(1) Points

Only one

As many as you want (*)

They must match the same number as the number of ELSE statements.

None; the command is ELSE IF;

Correct

24. What is the correct form of a simple IF statement? Mark for Review
(1) Points

IF condition THEN statement;

IF condition THEN statement;
END IF; (*)

IF condition;
THEN statement;
END IF;

IF condition
THEN statement
ENDIF;

Correct

25. What is the correct form of a compound IF statement? Mark for Review
(1) Points

IF condition
THEN statement1
ELSE statement 2;

IF condition
THEN statement1
ELSE statement 2;
END IF;

IF condition;
THEN statement1;
ELSE statement2;
END IF;

IF condition THEN statement1;
ELSE statement2;

```
END IF;
(*)
```

Correct

Section 5

26. Which of the following cursor attributes is set to the total number of rows returned so far? Mark for Review
(1) Points

%ISOPEN

%NOTFOUND

%FOUND

%ROWCOUNT (*)

Correct

27. Which of the following cursor attributes evaluates to TRUE if the cursor is open? Mark for Review
(1) Points

%ISOPEN (*)

%NOTFOUND

%FOUND

%ROWCOUNT

Correct

28. Assume that you have declared a cursor called C_EMP. Which of the following statements about C_EMP is correct? (Choose two.) Mark for Review
(1) Points

(Choose all correct answers)

You can use c_emp%NOTFOUND to exit a loop. (*)

You can fetch rows when c_emp%ISOPEN evaluates to FALSE.

You can use c_emp%ROWCOUNT to return the number of rows returned by the cursor so far. (*)

You can use c_emp%FOUND after the cursor is closed.

Correct

29. The employees table contains 11 columns. The following block declares a cursor and a record based on the cursor:

```
DECLARE
    CURSOR emp_curs IS
       SELECT * FROM employees;
    v_emp_rec emp_curs%ROWTYPE;
```

A twelfth column is now added to the employees table. Which of the following statements is true?

Mark for Review
(1) Points

The declaration of emp_rec must be changed to add an extra field.

The block will still work correctly without any changes to the PL/SQL code. (*)

The block will fail and an INVALID_CURSOR exception will be raised.

An extra scalar variable must be declared to correspond to the twelfth table column.

Correct

30. Which of the following statements about the %ISOPEN cursor attribute is true?

Mark for Review
(1) Points

You can issue the %ISOPEN cursor attribute only when a cursor is open.

You can issue the %ISOPEN cursor attribute only when more than one record is returned.

You can issue the %ISOPEN cursor attribute when a cursor is open or closed. (*)

If a cursor is open, then the value of %ISOPEN is false.

Correct

Test: Mid Term Exam Semester 1 - Part II

Review your answers, feedback, and question scores below. An asterisk (*) indicates a correct answer.

The Mid Term Exam for Semester 1 is presented to you as two exams. This is Part II of the Mid Term Exam for Semester 1.

Section 5

31. The following cursor has been declared:

```
CURSOR emp_curs IS
   SELECT first_name, last_name, job_id, salary
   FROM employees;
```

Which of the following correctly declares a composite record with the same structure as the cursor?
Mark for Review
(1) Points

emp_rec emp_rec%ROWTYPE;

emp_rec emp_curs%TYPE;

emp_rec emp_curs%ROWTYPE; (*)

emp_rec cursor%ROWTYPE;

Correct

32. What is one of the advantages of using parameters with a cursor? Mark for Review
(1) Points

You can use a cursor FOR loop.

You can declare the cursor FOR UPDATE.

You do not need to DECLARE the cursor at all.

You can use a single cursor to fetch a different set of rows each time the cursor is opened. (*)

It will execute much faster than a cursor without parameters.

Correct

33. Examine the following declaration of a cursor with a parameter. What should be coded at Point A? DECLARE

```
CURSOR emp_curs(-- Point A --) IS
SELECT * FROM employees
WHERE job_id = p_job_id;
```

Mark for Review
(1) Points

p_job_id

ST_CLERK'

p_job_id VARCHAR2(25)

p_job_id VARCHAR2 (*)

job_id VARCHAR2

Correct

PlSQL feedback midterm semister 1 part2

34. Which of these is NOT a valid cursor declaration? Mark for Review
(1) Points

```
CURSOR emp_curs IS
SELECT salary
FROM employees
ORDER BY salary DESC;
```

```
CURSOR emp_curs IS
SELECT salary
FROM employees
WHERE last_name LIKE 'S%';
```

```
CURSOR emp_dept_curs IS
SELECT e.salary, d.department_name
FROM employees e, departments d
WHERE e.department_id = d.department_id;
```

```
CURSOR emp_curs IS
SELECT salary INTO v_salary
FROM employees;
```

(*)

Correct

35. The employees table contains 20 rows. What will happen when the following code is executed?

```
DECLARE
   CURSOR emp_curs IS
      SELECT job_id FROM employees;
    v_job_id employees.job_id%TYPE;
BEGIN
    OPEN emp_curs;
    LOOP
       FETCH emp_curs INTO v_job_id;
       DBMS_OUTPUT.PUT_LINE(v_job_id);
       EXIT WHEN emp_curs%NOTFOUND;
    END LOOP;
    CLOSE emp_curs;
END;
```

Mark for Review
(1) Points

20 job_ids will be displayed.

The block will fail and an error message will be displayed.

21 rows of output will be displayed; the first job_id will be displayed twice.

21 rows of output will be displayed; the last job_id will be displayed twice.

(*)

Correct

36. For which type of SQL statement must you use an explicit cursor? Mark for Review
(1) Points

DML statements that process more than one row.

Queries that return more than one row. (*)

Data Definition Language (DDL) statements.

Queries that return a single row.

Correct

37. An implicit cursor can be used for a multiple-row SELECT statement. True or False? Mark for Review
(1) Points

True

False (*)

Correct

38. Place the following statements in the correct sequence:

```
OPEN my_curs;
CLOSE my_curs;
CURSOR my_curs IS SELECT my_column FROM my_table;
FETCH my_curs INTO my_variable;
```

Mark for Review
(1) Points

C,D,A,B

C,A,D,B (*)

A,C,D,B

C,A,B,D

Correct

39. What will happen when the following code is executed?

```
DECLARE CURSOR emp_curs IS
    SELECT salary FROM employees;
    v_salary employees.salary%TYPE;
BEGIN
    OPEN emp_curs;
    FETCH emp_curs INTO v_salary;
    CLOSE emp_curs;
    FETCH emp_curs INTO v_salary;
END;
```

Mark for Review
(1) Points

The block will fail and an INVALID_CURSOR exception will be raised. (*)

The first employee row will be fetched twice.

The first two employee rows will be fetched.

The block will fail and a TOO_MANY_ROWS exception will be raised.

Correct

40. An explicit cursor must always be declared, opened and closed by the PL/SQL programmer. True or False? Mark for Review
(1) Points

True

False (*)

Incorrect. Refer to Section 5.

Test: Mid Term Exam Semester 1 - Part II

Review your answers, feedback, and question scores below. An asterisk (*) indicates a correct answer.

The Mid Term Exam for Semester 1 is presented to you as two exams. This is Part II of the Mid Term Exam for Semester 1.

Section 5

41. Examine the following code:

```
DECLARE
    CURSOR emp_curs IS
       SELECT last_name, salary
       FROM employees
       ORDER BY salary;
    v_last_name employees.last_name%TYPE;
    v_salary employees.salary%TYPE;
BEGIN
...
```

which of the following statements successfully opens the cursor and fetches the first row of the active set?
Mark for Review
(1) Points

```
    OPEN emp_curs;
FETCH emp_curs INTO v_last_name, v_salary;
```
(*)

```
    OPEN emp_curs;
FETCH emp_curs INTO v_salary, v_last_name;
```

```
    OPEN emp_curs;
FETCH FIRST emp_curs INTO v_last_name, v_salary;
```

```
    OPEN emp_curs;
FETCH emp_curs;
```

Correct

42. Consider the following cursor:

```
CURSOR c IS
    SELECT e.last_name, e.salary, d.department_name
    FROM employees e JOIN departments d
    USING(department_id)
    WHERE e.last_name='Smith'
    FOR UPDATE;
```

When the cursor is opened and rows are fetched, what is locked?
Mark for Review
(1) Points

The whole EMPLOYEES table is locked.

In the EMPLOYEES table, only the 'Smith' rows are locked. Nothing in the DEPARTMENTS table is locked.

Each 'Smith' row is locked and Smith's matching rows in DEPARTMENTS are locked. No other rows are locked in either table. (*)

The whole EMPLOYEES and DEPARTMENTS tables are locked.

Nothing is locked because the cursor was not declared with NOWAIT.

Correct

43. Examine the following code:

```
DECLARE
CURSOR c IS SELECT * FROM employees FOR UPDATE;
c_rec c%ROWTYPE;
BEGIN
OPEN c;
FOR i IN 1..20 LOOP
FETCH c INTO c_rec;
IF i = 6 THEN
UPDATE employees SET first_name = 'Joe'
WHERE CURRENT OF c;
END IF;
END LOOP;
CLOSE c;
END;
```

Which employee row or rows will be updated when this block is executed? Mark for Review

(1) Points

The first 6 fetched rows will be updated.

No rows will be updated because you locked the rows when the cursor was opened.

The 6th fetched row will be updated. (*)

The block will not compile because the cursor should have been declared .... FOR UPDATE WAIT 5;

None of the above.

Correct

44. User TOM has locked a row in the WORKERS table. Now, user DICK wants to open the following cursor:
CURSOR c IS
SELECT * FROM workers FOR UPDATE NOWAIT;
What will happen when DICK opens the cursor and tries to fetch rows? Mark for Review
(1) Points

TOM's session is rolled back. DICK's session successfully fetches rows from the cursor.

DICK's session waits indefinitely.

Both sessions wait for a few seconds; then the system breaks all locks and both sessions raise an exception.

DICK's session immediately raises an exception. (*)

The c%NOWAIT attribute is set to TRUE.

Correct

45. When using a cursor FOR loop, OPEN, CLOSE and FETCH statements should not be explicitly coded. True or False? Mark for Review
(1) Points

True (*)

False

Correct

46. There are no employees in department_id 75.
Which of the following is NOT a valid cursor FOR loop with a subquery? Mark for Review
(1) Points

FOR emp_rec IN
(SELECT last_name, salary FROM employees) LOOP ...

FOR emp_rec IN

(SELECT * FROM employees) LOOP ...

FOR emp_rec IN
(SELECT last_name, salary FROM employees ORDER BY last_name) LOOP ...

FOR emp_rec IN
(SELECT * FROM employees WHERE department_id = 75) LOOP ...

None of the above. They are all valid. (*)

Correct

47. The following code fragment shows a cursor FOR loop:
FOR emp_record IN emp_cursor LOOP ......

Which of the following do NOT need to be coded explicitly? (Choose three.)
Mark for Review
(1) Points

(Choose all correct answers)

OPEN emp_cursor; (*)

DECLARE CURSOR emp_cursor IS ...

emp_record emp_cursor%ROWTYPE; (*)

FETCH emp_cursor INTO emp_record; (*)

END LOOP;

Correct

48. You want to display each row from the DEPARTMENTS table, and immediately underneath it, a list of all EMPLOYEES in that department. Which of the following is a good way to do this? Mark for Review
(1) Points

Use a single cursor, declared as SELECT * FROM employees GROUP BY department_id;

Use two cursors, one for each of the two tables. Declare the EMPLOYEES cursor with a parameter for the DEPARTMENT_ID. (*)

Write a SELECT statement which JOINs the two tables, and use CONNECT BY PRIOR and LEVEL to display the rows in the correct order.

Use a single cursor with a cursor FOR loop.

Change the physical model so that all employee and department data is in a single table.

Correct

49. Which of the following is a good reason to declare and use multiple cursors in a single PL/SQL block? Mark for Review
(1) Points

Multiple cursors improve performance. They are faster than using a single cursor.

Multiple cursors use less memory than a single cursor.

Multiple cursors allow us to fetch rows from two or more related tables without using a JOIN. (*)

Multiple cursors are the only way to use cursors with parameters.

Multiple cursors can be opened many times, while a single cursor can be opened only once.

Correct

50. When using multiple nested cursors, what kinds of loops can you use? Mark for Review
(1) Points

Cursor FOR loops only.

Basic loops only.

WHILE loops only.

None of the above.

All of the above. (*)

Correct

1. PL/SQL extends SQL by including all of the following except: Mark for Review
(1) Points

variables

conditional statements

reusable program units

constants

nonprocedural constructs (*)

Correct

2. SQL is a common access language for many types of databases, including Oracle. True or False? Mark for Review
(1) Points

True (*)

False

Correct

3. A program which specifies a list of operations to be performed sequentially to achieve the desired result can be called: Mark for Review
(1) Points

declarative

nondeclarative

procedural (*)

low level

Correct

4. Which statements are optional in a PL/SQL block? (Choose two.) Mark for Review
(1) Points

(Choose all correct answers)

DECLARE (*)

BEGIN

EXCEPTION (*)

END;

Correct

5. Which PL/SQL block type must return a value? Mark for Review
(1) Points

Anonymous

Function (*)

Procedure

Correct

6. Which keywords must be included in every PL/SQL block? (Choose two.) Mark for Review
(1) Points

(Choose all correct answers)

DECLARE

END; (*)

EXCEPTION

BEGIN (*)

DBMS_OUTPUT.PUT_LINE

Correct

7. Given below are the parts of a PL/SQL block:
1. END;
2. EXCEPTION
3. DECLARE
4. BEGIN

Arrange the parts in order.
Mark for Review
(1) Points

2,1,4,3

3,4,2,1 (*)

3,2,4,1

4,3,2,1

Correct

8. Which lines of code will correctly display the message "The cat sat on the mat"? (Choose two.) Mark for Review
(1) Points

(Choose all correct answers)

DBMS_OUTPUT.PUT_LINE('The cat sat on the mat'); (*)

DBMS_OUTPUT.PUT_LINE(The cat sat on the mat);

DBMS_OUTPUT.PUT_LINE('The cat' || 'sat on the mat');

DBMS_OUTPUT.PUT_LINE('The cat sat ' || 'on the mat'); (*)

Correct

9. Which of the following tools can NOT be used to develop and test PL/SQL code?
Mark for Review
(1) Points

Oracle Jdeveloper

Oracle Application Express

Oracle JSQL (*)

Oracle iSQL*Plus

Correct

10. What is the purpose of using DBMS_OUTPUT.PUT_LINE in a PL/SQL block? Mark for Review
(1) Points

To perform conditional tests

To allow a set of statements to be executed repeatedly

To display results to check if our code is working correctly (*)

To store new rows in the database

Correct

11. PL/SQL can be used not only with an Oracle database, but also with any kind of relational database. True or False? Mark for Review
(1) Points

True

False (*)

Correct

12. Which of the following statements about exception handling in PL/SQL is false? Mark for Review
(1) Points

You can prepare for database exceptions by creating exception handlers.

You can prepare for application exceptions by creating exception handlers.

Exception handling code tells your program what to do when an error is encountered.

Exception handling code can be grouped together in a PL/SQL block.

None of the above (*)

Correct

13. Which of the following can you use PL/SQL to do? Mark for Review
(1) Points

Update data (DML)

Develop Web applications using the Web Application Toolkit

Manage database security

Create customized reports

All of the above (*)

Correct

Section 2

14. Which of these are PL/SQL data types? (Choose three.) Mark for Review (1) Points

(Choose all correct answers)

Scalar (*)

Identifier

Delimiter

Composite (*)

LOB (*)

Correct

15. A movie is an example of which category of data type? Mark for Review (1) Points

Scalar

Composite

Reference

LOB (*)

Correct

16. _____ are meant to store large amounts of data. Mark for Review (1) Points

Variables

Scalar data types

LOBs (*)

Correct

17. When a variable is defined using the CONSTANT keyword, the value of the variable cannot change. True or False? Mark for Review (1) Points

True (*)

False

Correct

18. Identify which of the following assignment statements are valid. (Choose three.) Mark for Review
(1) Points

(Choose all correct answers)

v_last_name := Chandra;

v_blackout_date := '31-DEC-2006'; (*)

v_population := 333444; (*)

v_music_type := 'ROCK'; (*)

Incorrect. Refer to Section 2.

19. When a variable is defined using the NOT NULL keywords, the variable must contain a value. True or False? Mark for Review
(1) Points

True (*)

False

Correct

20. Assignment statements can continue over several lines in PL/SQL. True or False? Mark for Review
(1) Points

True (*)

False

Correct

21. Evaluate the following declaration. Determine whether or not it is legal.
DECLARE
maxsalary NUMBER(7) = 5000;
Mark for Review
(1) Points

Correct.

Not correct. (*)

Correct

22. Reserved words can be used as identifiers. True or False? Mark for Review
(1) Points

True

False (*)

Incorrect. Refer to Section 2.

23. Which of the following are valid identifiers? (Choose two.) Mark for Review (1) Points

(Choose all correct answers)

yesterday (*)

yesterday's date

number_of_students_in_the_class

v$testresult (*)

#students

Correct

24. Which of the following are valid identifiers? (Choose two.) Mark for Review (1) Points

(Choose all correct answers)

Full Name

students_street_address (*)

v_code (*)

#hours

completion_%

Correct

25. When an exception occurs within a PL/SQL block, the remaining statements in the executable section of the block are skipped. True or False? Mark for Review (1) Points

True (*)

False

Correct

26. Examine the following code. Line A causes an exception. What will be displayed when the block is executed?

```
DECLARE
    var_a NUMBER := 6;
    var_b DATE;
```

```
BEGIN
    var_a := var_a * 2;
    var_b := '28 December 2006'; -- Line A
    var_a := var_a * 2;
EXCEPTION
    WHEN OTHERS THEN
       DBMS_OUTPUT.PUT_LINE(var_a);
END;
```

Mark for Review
(1) Points

12 (*)

24

6

Nothing will be displayed

Correct

27. Examine the following code. At Line A, we want to assign a value of 22 to the outer block's variable v_myvar. What code should we write at Line A?

```
<<outer_block>>
DECLARE
    v_myvar NUMBER;
BEGIN
    <<inner_block>>
    DECLARE
       v_myvar NUMBER := 15;
    BEGIN
       -- Line A
    END;
END;
```

Mark for Review
(1) Points

outer_block.v_myvar := 22; (*)

v_myvar := 22;

<<outer_block>>.v_myvar := 22;

v_myvar(outer_block) := 22;

We cannot reference the outer block's variable because both variables have the same name

Correct

28. When nested blocks are used, which blocks can or must be labeled? Mark for Review
(1) Points

The inner block must be labeled, the outer block can be labeled.

Both blocks must be labeled

Nested blocks cannot be labeled

The outer block must be labeled if it is to be referred to in the inner block. (*)

Correct

29. What will be displayed when the following code is executed?

```
DECLARE
    varA NUMBER := 12;
BEGIN
    DECLARE
       varB NUMBER := 8;
    BEGIN
       varA := varA + varB;
    END;
    DBMS_OUTPUT.PUT_LINE(varB);
END;
```

Mark for Review
(1) Points

8

12

Nothing, the block will fail with an error (*)

20

VarB

Correct

30. Examine the following code. What is the final value of V_MYVAR ?

```
DECLARE
    v_myvar NUMBER;
BEGIN
    v_myvar := 1 + 2 * 3;
    v_myvar := v_myvar * 2;
END;
```

Mark for Review
(1) Points

81

49

14 (*)

18

Correct

You can create a Web site application written entirely in PL/SQL. True or False? Mark for Review

PlSQL feedback midterm semister 1 part2

(1) Points

True (*)

False

Which of the following can be done using PL/SQL? Mark for Review

(1) Points

Create complex applications.

Retrieve and modify data in Oracle database tables.

Manage database tasks such as security.

Create custom reports.

All of the above (*)

When multiple SQL statements are combined into PL/SQL blocks, performance improves. True or False? Mark for Review

(1) Points

True (*)

False

PL/SQL differs from C and Java in which of the following ways? (Choose two.) Mark for Review

(1) Points

(Choose all correct answers)

It requires an Oracle database or tool. (*)

It does not support object-oriented programming.

It is the most efficient language to use with an Oracle database. (*)

It is the most complex programming language to learn.

It is not portable to other operating systems

Which of the following can be compiled as a standalone program outside the database? Mark for Review

(1) Points

A program developed in PL/SQL

A program developed in Java

A program developed in C

All the above

Programs developed in Java or C, but not in PL/SQL. (*)

Procedural constructs give you better control of your SQL statements and their execution. True or False? Mark for Review
(1) Points

True (*)

False

You can create a Web site application written entirely in PL/SQL. True or False? Mark for Review
(1) Points

True (*)

False

PL/SQL differs from C and Java in which of the following ways? (Choose two.) Mark for Review
(1) Points

(Choose all correct answers)

It requires an Oracle database or tool. (*)

It does not support object-oriented programming.

It is the most efficient language to use with an Oracle database. (*)

It is the most complex programming language to learn.

It is not portable to other operating systems.

Which of the following can be done using PL/SQL? Mark for Review
(1) Points

Create complex applications.

Retrieve and modify data in Oracle database tables.

Manage database tasks such as security.

Create custom reports.
All of the above (*)
When multiple SQL statements are combined into PL/SQL blocks, performance improves. True or False? Mark for Review
(1) Points

True (*)

False

Procedural constructs give you better control of your SQL statements and their execution. True or False? Mark for Review
(1) Points

True (*)

False

Incorrect Incorrect. Refer to Section 1.
Which of the following can be compiled as a standalone program outside the database? Mark for Review
(1) Points

A program developed in PL/SQL

A program developed in Java

A program developed in C

All the above

Programs developed in Java or C, but not in PL/SQL. (*)

How can you display results to check that a PL/SQL block is working correctly? Mark for Review
(1) Points

You don't need to do anything, the results will display automatically.

Use an Exception section

Use DBMS_OUTPUT.PUT_LINE (*)

PlSQL feedback midterm semister 1 part2

Write a C or Java program to display the results

Which statements are mandatory in a PL/SQL block? (Choose two.) Mark for Review
(1) Points

(Choose all correct answers)

DECLARE

BEGIN (*)

EXCEPTION

END; (*)

What are the characteristics of an anonymous block? (Choose two.) Mark for Review
(1) Points

(Choose all correct answers)

Unamed (*)

Stored in the database

Compiled each time the application is executed (*)

Can be declared as procedures or as functions

What are the characteristics of a PL/SQL stored subprogram? (Choose two.) Mark for Review
(1) Points

(Choose all correct answers)

Named (*)

Not stored in the database

Can be invoked at any time (*)

Do not exist after they are executed

Which of the following is NOT a PL/SQL programming environment? Mark for Review
(1) Points

PlSQL feedback midterm semister 1 part2

Oracle jDeveloper

SQL*Plus

gSQL*Plus (*)

SQL Workshop in Application Express

Incorrect Incorrect. Refer to Section

What is wrong with this PL/SQL anonymous block?

```
BEGIN
    DBMS_OUTPUT.PUT_LINE('Hello');
    DBMS_OUTPUT.PUT_LINE(' and Goodbye');
```

Mark for Review
(1) Points

The Declaration section is missing

The Exception section is missing

There is nothing wrong with the block, it will work fine.

The END; statement is missing (*)

n a PL/SQL block, which of the following should not be followed by a semicolon?
Mark for Review
(1) Points

DECLARE (*)

END

All SQL statements

All PL/SQL statements

Which sections of a PL/SQL block are optional? Mark for Review
(1) Points

Declaration and Executable

Declaration and Exception (*)

Exception only

Executable only

Which lines of code will correctly display the message "Hello World" ? (Choose two.)
Mark for Review
(1) Points

(Choose all correct answers)

DBMS_OUTPUT('Hello World');

DBMS_OUTPUT.PUT_LINE('Hello World'); (*)

DBMS_OUTPUT.PUT_LINE('Hello' || 'World');

DBMS_OUTPUT.PUT_LINE('Hello' || ' ' || 'World'); (*)

Which of the following is a PL/SQL programming environment? Mark for Review
(1) Points

Oracle Cdeveloper

Java*Plus

PL/SQL Express

SQL*Workshop in Application Express (*)

What can you use to change the column heading of calculated values in a SQL statement? Mark for Review
(1) Points

Multiplication operator

Column alias (*)

Concatenation operator

The DISTINCT keyword

If you want to SELECT all the columns of data in a table, you use which of the following symbols? Mark for Review
(1) Points

&

%

$

* (*)

he concatenation operator ... Mark for Review
(1) Points

Brings columns or character strings together

Creates a resultant column that is a character expression

Is represented by two vertical bars (||)

All of the above (*)

Which statement would display the departments in the EMPLOYEES table without displaying any duplicates? Mark for Review
(1) Points

SELECT ALL department_id
FROM employees;

SELECT department_id
FROM employees;

SELECT department_id
FROM employees
having ROWID=1;

SELECT DISTINCT department_id
FROM employees;

(*)

Which of the following statements lists each employee's employee_id, salary, and salary plus a 20 percent bonus? Mark for Review
(1) Points

SELECT emp_id, salary, salary*.2
FROM employees;

SELECT emp_id, salary, salary*1.2
FROM employees;

(*)

SELECT emp_id, salary, salary*.8
FROM employees;

SELECT emp_id, salary, salary*20
FROM employees;

What SQL statement will return the ID, name, and area of all countries in the WF_COUNTRIES table, listed in order of greatest area to least area? Mark for Review
(1) Points

SELECT country_id, country_name, area
FROM wf_countries
ORDER BY area DESC;

(*)

SELECT country_id, country_name, area
FROM wf_countries
ORDER BY area ASC;

SELECT country_id, country_name, area
FROM wf_countries
ORDER BY country_name;

SELECT country_id, country_name, area
FROM wf_countries
GROUP BY area; pr />

Which statement would select salaries that are greater than or equal to 2500 and less than or equal to 3500? Choose two correct answers. Mark for Review
(1) Points

(Choose all correct answers)

WHERE salary >= 2500 AND salary <= 3500 (*)

WHERE salary <=2500 AND salary >= 3500

WHERE salary BETWEEN 2500 AND 3500 (*)

WHERE BETWEEN salary = 2500 AND salary = 3500

When using the LIKE operator, the "%" and "_" symbols can be used to do a pattern-matching, wild card search. True or False? Mark for Review
(1) Points

True (*)

False

Examine the following statement:

```
SELECT country_name, population, population*.01
FROM wf_countries;
```

How would you modify this statement to display "Country", "Population", and "Expected Growth" as the column headings?
Mark for Review
(1) Points

SELECT country_name "COUNTRY", population "POPULATION", population*.01 "EXPECTED GROWTH"
FROM wf_countries;

(*)

SELECT country_name COUNTRY, population POPULATION, population*.01 EXPECTED GROWTH
FROM wf_countries;

SELECT country_name 'COUNTRY', population 'POPULATION', population*.01 'EXPECTED GROWTH'
FROM wf_countries;

SELECT country_name, population, population*.01
FROM wf_countries
AS "COUNTRY", "POPULATION", "EXPECTED GROWTH";

The F_FOOD_ITEMS table contains the FOOD_ITEM_NUMBER and the REGULAR_CODE columns. Which statement would display the FOOD_ITEM_NUMBER joined with the REGULAR_CODE without any space in between them? Mark for Review
(1) Points

SELECT food_item_number ' ' regular_code
FROM f_food_items;

SELECT food_item_number UNION regular_code
FROM f_food_items;

SELECT food_item_number || regular_code
FROM f_food_items;

(*)

```
        SELECT food_item_numberregularcode
FROM f_food_items;
```

Which of the following statements will display a sentence such as the following:
Aruba has an area of 193.
for every country in the WF_COUNTRIES table? Mark for Review
(1) Points

```
        SELECT country_name || ' has an area of ' || area
FROM wf_countries;
```

```
        SELECT country_name || 'has an area of' || area
FROM wf_countries;
```

```
        SELECT country_name || ' has an area of ' || area || '.'
FROM wf_countries;
```

(*)

```
        SELECT country_name " has an area of " area "."
FROM wf_countries;
```

Which of the following statements will generate a sentence such as the following:
The national holiday for United Arab Emirates is Independence Day.
for every country in the WF_COUNTRIES table?
        Mark for Review
(1) Points

```
        SELECT 'The national holiday for '|| country_name || ' is ' ||
national_holiday_name
FROM wf_countries;
```

```
        SELECT "The national holiday for "|| country_name || " is " ||
national_holiday_name || "."
FROM wf_countries;
```

```
        SELECT 'The national holiday for '|| country_name || ' is ' ||
national_holiday_name || '.'
FROM wf_countries;
```

(*)

```
        SELECT 'The national holiday for || country_name || is ||
national_holiday_name || .'
FROM wf_countries;
```

Which of the following statements diplays the population of the Republic of Benin
(country_id 229) after a 3 percent growth in its population? Mark for Review

(1) Points

SELECT country_name, population*.03
FROM wf_countries
WHERE country_id=229;

SELECT country_name, population*1.03
FROM wf_countries
WHERE country_id=229;

(*)

SELECT country_name, population*30
FROM wf_countries
WHERE country_id=229;

SELECT country_name, population+population*.3
FROM wf_countries
WHERE country_id=229;

Which of the following is not a number function? Mark for Review
(1) Points

TO_DATE (*)

ROUND

MOD

TRUNC

The following SQL statement will display the value: 456. True or False?

```
SELECT TRUNC(ROUND(456.98))
FROM dual;
```

Mark for Review
(1) Points

True

False (*)

Which statement returns a user password combining the ID of an employee and the first 4 characters of their last name? Mark for Review
(1) Points

```
        SELECT CONCAT (employee_id, SUBSTR(last_name,4,1))
AS "User Passwords"
FROM employees;
```

```
        SELECT CONCAT (employee_id, INSTR(last_name,4,1))
AS "User Passwords"
FROM employees;
```

```
        SELECT CONCAT (employee_id, INSTR(last_name,1,4))
AS "User Passwords"
FROM employees;
```

```
        SELECT CONCAT (employee_id, SUBSTR(last_name,1,4))
AS "User Passwords"
FROM employees;
```

(*)

Which query would return a whole number if today's date is 26-MAY-04? Mark for Review
(1) Points

```
        SELECT TRUNC(MONTHS_BETWEEN(SYSDATE,'19-MAR-79') /12)
AS YEARS
FROM DUAL;
```

(*)

```
        SELECT TRUNC(YEARS_BETWEEN(SYSDATE,'19-MAR-79') /12)
AS YEARS
FROM DUAL;
```

```
        SELECT MONTHS_BETWEEN(SYSDATE,'19-MAR-79') /12
AS YEARS
FROM DUAL;
```

None of the above

Which function compares two expressions? Mark for Review
(1) Points

NVL

NULLIF (*)

NVL2

NULL

Assume that today is December 31, 2007. What would be the output of the following statement?

```
SELECT TO_CHAR(SYSDATE, 'DD/MM/Y') FROM DUAL;
```

Mark for Review
(1) Points

12/31/7

31-12-07

31/12/2007

31/12/7 (*)

Assume that today is January 10, 2008. What would be the output of the following statement?

```
SELECT TO_CHAR(SYSDATE, 'ddth "of" Month, YYYY') FROM DUAL;
```

Mark for Review
(1) Points

10th of January, 2008 (*)

10 January, 2008

10-January-2008

January 10th, 2008

What is returned by the following statement?

```
SELECT CONCAT('Today is','Thursday!') FROM DUAL;
```

Mark for Review
(1) Points

TodayisThursday!

Today isThursday! (*)

today is thursday!

Today is Thursday!

What does the following SQL SELECT statement return?

```
SELECT UPPER( SUBSTR('Database Programming', INSTR('Database Programming','P'),20))
```

```
FROM dual;
```

Mark for Review
(1) Points

Programming

PROGRAMMING (*)

Database

DATABASE

What function would you use to return the highest date in a month? Mark for Review
(1) Points

FINAL_DAY

END_DAY

HIGHEST_DAY

LAST_DAY (*)

Which SQL statement will display each country's name with the first letter (only) of each word in uppercase? Mark for Review
(1) Points

```
SELECT UPPER(country_name)
FROM wf_countries;
```

```
SELECT lower(country_name)
FROM wf_countries;
```

```
SELECT INITCAP(country_name)
FROM wf_countries;
```

(*)

```
SELECT country_name
FROM wf_countries
ORDER BY INITCAP(country_name);
```

NULL means the same thing as a space or 0 (zero). True or False? Mark for Review
(1) Points

True

False (*)

Constants must be initialized. True or False? Mark for Review
(1) Points

True (*)

False

After they are declared, variables can be used only once in an application. True or False? Mark for Review
(1) Points

True

False (*)

Examine the following variable declarations:
DECLARE v_number NUMBER := 10; v_result NUMBER;
Which of the following correctly assigns the value 50 to V_RESULT? Mark for Review
(1) Points

v_result := v_number * 5;

v_result := 100 / 2;

v_result := ROUND(49.77);

All of the above. (*)

Evaluate the following declaration. Determine whether or not it is legal.

```
DECLARE
    name,dept VARCHAR2(14);
```
Mark for Review
(1) Points

legal

illegal (*)

Evaluate the following declaration. Determine whether or not it is legal.

```
DECLARE
    test NUMBER(5);
```
Mark for Review
(1) Points

legal (*)

illegal

Which of the following are required when declaring a variable? (Choose two.)
Mark for Review
(1) Points

(Choose all correct answers)

Identifier name (*)

CONSTANT

Data type (*)

NOT NULL

A function called FORMAT_TODAYS_DATE accepts no parameters and returns today's date in the format: Month DD, YYYY
The following anonymous block invokes the function:

DECLARE v_today DATE; BEGIN -- invoke the function here

Which of the following statements correctly assigns the date variable v_today to the value returned by the format_todays_date function?
Mark for Review
(1) Points

format_todays_date := v_today('Month DD, YYYY');

v_today := format_todays_date ('Month DD, YYYY');

v_today := format_todays_date(v_today);

v_today := TO_DATE(format_todays_date, 'Month DD, YYYY'); (*)

The name of a variable is an example of an identifier. True or False? Mark for Review
(1) Points

True (*)

False

Which of the following is a valid naming convention for an identifier? (Choose two.) Mark for Review
(1) Points

(Choose all correct answers)

Can include letters or numbers (*)

Cannot contain a reserved word (*)

Can be over 30 characters

Can start with a number or special character

Which of the following are lexical units? (Choose two.) Mark for Review
(1) Points

(Choose all correct answers)

Data types

PL/SQL blocks

Identifiers (*)

Literals (*)

What characters must enclose non-numeric literal values? Mark for Review
(1) Points

Double quotes: " "

Parentheses: ()

Single quotes: ' ' (*)

What is a lexical unit? Mark for Review
(1) Points

A data type for a column

A building block of a PL/SQL block (*)

A type of variable

Which of the following symbols can be used to enclose a comment in PL/SQL? Mark for Review
(1) Points

? ?

*/ / *

:: ::

/* */ (*)

What are the data types of the variables in the following declaration?

```
DECLARE
fname VARCHAR2(20);
fname VARCHAR2(15) DEFAULT 'fernandez';
BEGIN
...
```

Mark for Review
(1) Points

Scalar (*)

Composite

LOB

A Scalar data type holds a ____ value. Mark for Review
(1) Points

Multi

Large

Single (*)

Which of the folowing are scalar data types? (Choose three.) Mark for Review
(1) Points

(Choose all correct answers)

Array

Character (*)

Table

Date (*)

Boolean (*)

PlSQL feedback midterm semister 1 part2

Which of the following is a composite data type? Mark for Review
(1) Points

CLOB

VARCHAR2

RECORD (*)

DATE

Which of the following are PL/SQL data types? (Choose three.) Mark for Review
(1) Points

(Choose all correct answers)

Large Objects (LOB) (*)

Lexical

Scalar (*)

Delimiter

Composite (*)

datatype specifies and restricts the possible data values that can be assigned to a variable. True or False? Mark for Review
(1) Points

True (*)

False

If you use the %TYPE attribute, you can avoid hard-coding the column name. True or False? Mark for Review
(1) Points

True

False (*)

Which of the following is NOT a character data type? Mark for Review
(1) Points

VARCHAR2

BOOLEAN (*)

CHAR

LONG

When declared using %TYPE, a variable will inherit ____ from the column on which it is based. Mark for Review
(1) Points

The name of the column

The value of the column

The data type and size of the column (*)

Code is easier to read if you declare one identifier per line. True or False? Mark for Review
(1) Points

True (*)

False

Which of the following is NOT a good guideline for declaring variables? Mark for Review
(1) Points

Declare one identifier per line

Use column names as identifiers (*)

Use NOT NULL when the variable must have a value

Which of the following variable declarations does NOT use a number data type? Mark for Review
(1) Points

v_count PLS_INTEGER := 0;

v_median_age NUMBER(6,2);

v_students LONG; (*)

v_count BINARY_INTEGER;

What kind of join is used in the following example?

```
SELECT e.employee_id, e.last_name, j.grade_level
FROM employees e, job_grades j
WHERE e.salary BETWEEN j.lowest_sal and j.highest_sal;
```

Mark for Review
(1) Points

Simple join

Equijoin

Nonequijoin (*)

Outer join

What does the following statement return?

```
SELECT e.last_name, d.department_id, d.department_name
FROM employees e, departments d
WHERE e.department_id(+) = d.department_id
ORDER BY e.department_id;
```

Mark for Review
(1) Points

Returns all departments, even if there are no employees in the department. (*)

Returns all employees, even if they have not been assigned to a department.

Returns only those departments that contain at least one employee

Returns all possible combinations of employees and departments.

A nonequijoin combines tables that have one or more exact matching columns. True or False? Mark for Review
(1) Points

True

False (*)

Table aliases can be used to shorten the syntax in join statements. True or False? Mark for Review
(1) Points

True (*)

False

What type of join returns rows for one table even when there are no matching rows in the other table? Mark for Review
(1) Points

Simple join

Equijoin

Nonequijoin

Outer join (*)

If table A has 20 rows and table B has 10 rows, how many rows will be returned if you perform a Cartesian product on those two tables? Mark for Review
(1) Points

20

10

200 (*)

120

Will the following statement execute correctly?

```
SELECT department_id, department_name, last_name
FROM employees e, departments d
WHERE e.department_id = d.department_id;
```

Mark for Review
(1) Points

Yes, there are no errors in this statement.

No, because one column has been ambiguously defined. (*)

No, because every column must be prefixed by its table alias, for example: e.last_name.

Yes, Oracle will resolve which department_id column comes from which table.

When a join condition is omitted completely the result is a Cartesian product in which all combinations of rows will be displayed. True or False? Mark for Review
(1) Points

True (*)

False

Which of the following SQL statements will display the name and a total of people with the same last name? Mark for Review
(1) Points

```
SELECT last_name, COUNT(employee_id)
FROM EMPLOYEES
GROUP BY last_name;
```

(*)

```
SELECT employee_id, COUNT(last_name)
FROM EMPLOYEES
GROUP BY last_name;
```

```
SELECT last_name, DISTINCT COUNT(employee_id)
FROM EMPLOYEES
GROUP BY last_name;
```

```
SELECT employee_id, DISTINCT(last_name)
FROM EMPLOYEES
GROUP BY last_name;
```

Single row subqueries may NOT include which of these operators? Mark for Review
(1) Points

ALL (*)

=

<>

>

When using a subquery, the =ANY and IN operators are logically identical; they will always give the same result as each other. True or False? Mark for Review
(1) Points

True (*)

False

The following EMPLOYEE_ID, SALARY, and COMMISSION_PCT data in the EMPLOYEES table for six employees.

```
DATA:   143, 2600, null
        144, 2500, null
        149, 10500, .2
        174, 11000, .3
        176, 8600, .2
        178, 7000, .15
```

What is the result of the following statement:

```
SELECT AVG(commission_pct)
FROM employees
WHERE employee_id IN( 143,144,149,174,176,178)
```

Mark for Review
(1) Points

0.1416

0.2125 (*)

The statement will fail because you cannot use more than one group function in a single statement.

0.2521

What would the following SQL statement return?

```
SELECT MAX(hire_date) FROM employees;
```

Mark for Review
(1) Points

The hire date of the longest serving employee.

The hire date of the newest (most recently hired) employee. (*)

The hire dates of all employees in ascending order.

The hire dates of all employees.

Read the following SELECT statement. Choose the column or columns that MUST be included in the GROUP BY clause.

```
SELECT region_id, COUNT(country_id)
FROM wf_countries
GROUP BY ?????
```

Mark for Review
(1) Points

region_id, COUNT(country_id)

region_id,country_id

country_id

region_id (*)

Group functions cannot be used in subqueries because they contain too many rows.
True or False? Mark for Review
(1) Points

True

False (*)

What will be returned when the following statement is executed?

```
SELECT last_name
FROM employees
WHERE salary > ALL
(SELECT salary FROM employees
WHERE job_id = 'IT_PROG');
```

Mark for Review
(1) Points

The names of all IT Programmers.

The names of employees who earn more than every IT Programmer. (*)

The names of employees who earn more than at least one IT Programmer.

The names of employees who earn more than half of the IT Programmers.

Which of the following statements about implicit conversions is NOT true? Mark for Review
(1) Points

Code containing implicit conversions typically runs faster than code containing explicit conversions. (*)

Code containing implicit conversions may not work in the future if Oracle changes the conversion rules.

Code containing implicit conversions is harder to read and understand.

PL/SQL statements must be written on a single line.
Mark for Review
(1) Points

True

False (*)

Which of the following are valid PL/SQL operators? (Choose three.) Mark for Review
(1) Points

(Choose all correct answers)

Concatenation (*)

Exception

Exponential (*)

Arithmetic (*)

Which of the following data type conversions can be done implicitly? (Choose two.) Mark for Review
(1) Points

(Choose all correct answers)

DATE to NUMBER

NUMBER to VARCHAR2 (*)

NUMBER to PLS_INTEGER (*

The LENGTH and ROUND functions can be used in PL/SQL statements. True or False? Mark for Review
(1) Points

True (*)

False

Examine the following code: DECLARE x VARCHAR2(20); BEGIN x:= 5 + 4 * 5 ; DBMS_OUTPUT.PUT_LINE(x); END; What value of x will be displayed? Mark for Review
(1) Points

45

29

25 (*)

14

What will happen when the following code is executed?
```
DECLARE v_new_date DATE;
BEGIN
v_new_date := 'Today';
DBMS_OUTPUT.PUT_LINE(v_new_date);
END;
```
Mark for Review
(1) Points

The block will execute and display today's date.

The block will execute and display the word "Today".

The block will fail because the character value "Today" cannot be implicitly converted to a date. (*)

Incorrect Incorrect. Refer to Section 2.

Which explicit function is used to convert a character into a number? Mark for Review
(1) Points

TO_DATE

TO_NUMBER (*)

TO_CHAR

PL/SQL can implicitly convert a CHAR to a NUMBER, provided the CHAR contains a numeric value, for example '123'. True or False? Mark for Review
(1) Points

True (*)

False

The DECODE and MAX functions can be used in PL/SQL statements. True or False? Mark for Review
(1) Points

True

False (*)

Using implicit conversions is good programming practice. Mark for Review
(1) Points

True

False (*)

Examine the following block. What should be coded at Line A?
DECLARE
v_char VARCHAR2(8) := '24/09/07';
v_date DATE;
BEGIN
v_date := ..... Line A
END;
Mark for Review
(1) Points

v_date := FROM_CHAR(v_char,'dd/mm/yy');

v_date := TO_DATE(v_char,'dd/mm/yy'); (*)

v_date := v_char;

When PL/SQL converts data automatically from one data type to another, it is called _____ conversion. Mark for Review
(1) Points

Explicit

Implicit (*)

TO_CHAR

The TO_CHAR function is used for explicit data type conversions. True or False?
Mark for Review
(1) Points

True (*)

False

Which of the following is correct? Mark for Review
(1) Points

v_family_name = SMITH;

V_FAMILY_NAME = SMITH;

v_family_name := SMITH;

v_family_name := 'SMITH'; (*)

Using implicit conversions is good programming practice. Mark for Review
(1) Points

True

False (*)

Examine the following code: DECLARE x VARCHAR2(20); BEGIN x:= 5 + 4 * 5 ; DBMS_OUTPUT.PUT_LINE(x); END; What value of x will be displayed? Mark for Review
(1) Points

45

29

25 (*)

14

The DECODE and MAX functions can be used in PL/SQL statements. True or False? Mark for Review
(1) Points

True

False (*)

Which of the following are valid PL/SQL operators? (Choose three.) Mark for Review
(1) Points

(Choose all correct answers)

Concatenation (*)

Exception

Exponential (*)

Arithmetic (*)

Which of the following statements about implicit conversions is NOT true? Mark for Review
(1) Points

Code containing implicit conversions typically runs faster than code containing explicit conversions. (*)

Code containing implicit conversions may not work in the future if Oracle changes the conversion rules.

Code containing implicit conversions is harder to read and understand.

Which explicit function is used to convert a character into a number? Mark for Review
(1) Points

TO_DATE

TO_NUMBER (*)

TO_CHAR

Examine the following block. What should be coded at Line A?

```
DECLARE
v_char VARCHAR2(8) := '24/09/07';
v_date DATE;
BEGIN
v_date := ....... Line A
END;
```

Mark for Review
(1) Points

v_date := FROM_CHAR(v_char,'dd/mm/yy');

v_date := TO_DATE(v_char,'dd/mm/yy'); (*)

v_date := v_char;

The TO_CHAR function is used for explicit data type conversions. True or False? Mark for Review
(1) Points

True (*)

False

Incorrect Incorrect. R

The LENGTH and ROUND functions can be used in PL/SQL statements. True or False? Mark for Review

PlSQL feedback midterm semister 1 part2

(1) Points

True (*)

False

Incorrect Incorrect. Refer to Sectio

PL/SQL statements must be written on a single line. Mark for Review
(1) Points

True

False (*

Which of the following data type conversions can be done implicitly? (Choose two.)
Mark for Review
(1) Points

(Choose all correct answers)

DATE to NUMBER

NUMBER to VARCHAR2 (*)

NUMBER to PLS_INTEGER (*)

When PL/SQL converts data automatically from one data type to another, it is called _____ conversion. Mark for Review
(1) Points

Explicit

Implicit (*)

TO_CHAR

What will happen when the following code is executed?

```
DECLARE v_new_date DATE;
BEGIN
v_new_date := 'Today';
DBMS_OUTPUT.PUT_LINE(v_new_date);
END;
```

Mark for Review
(1) Points

The block will execute and display today's date.

The block will execute and display the word "Today".

The block will fail because the character value "Today" cannot be implicitly converted to a date. (*)

PL/SQL can implicitly convert a CHAR to a NUMBER, provided the CHAR contains a numeric value, for example '123'. True or False? Mark for Review
(1) Points

True (*)

False

Which of the following is correct? Mark for Review
(1) Points

v_family_name = SMITH;

V_FAMILY_NAME = SMITH;

v_family_name := SMITH;

v_family_name := 'SMITH'; (*)

What happens when an exception occurs in the executable section of a PL/SQL block? Mark for Review
(1) Points

Oracle keeps trying to re-execute the statement which caused the exception.

The remaining statements in the executable section are not executed. Instead, Oracle looks for an EXCEPTION section in the block. (*)

The remaining statements in the executable section of the block are executed.

The exception is always propagated to the calling environment.

An inner block is nested within an outer block. An exception occurs within the inner block, but the inner block does not have an EXCEPTION section. What happens? Mark for Review
(1) Points

The exception is propagated to the outer block and the remaining executable statements in the outer block are skipped. (*)

The exception is propagated to the outer block and the remaining executable statements in the outer block are executed.

Oracle automatically tries to re-execute the inner block.

The outer block is bypassed and the exception is always propagated to the calling environment.

What is wrong with this code?

```
DECLARE
    v_a NUMBER;
BEGIN
    v_a := 27;
    <<inner_block>>
    BEGIN
       v_a := 15;
END;
```

Mark for Review
(1) Points

The outer block has no label.

Variable v_a is out of scope within the inner block and therefore cannot be referenced.

The inner block has no END; statement. (*)

Nothing is wrong, the code will execute successfully.

Examine the following code. At Line A, we want to assign a value of 25 to the outer block's variable (V1). What must we do?

```
DECLARE
    v_myvar NUMBER; -- This is V1
BEGIN
    DECLARE
       v_myvar NUMBER := 8;
       BEGIN
          -- Line A
       END;
END;
```

Mark for Review
(1) Points

At Line A, code:
v_myvar := 25;

Label both blocks and at line A, code:
v_myvar := 25;

It cannot be done because the outer block's v_myvar is out of scope at Line A.

Label the outer block and (at Line A) dot-prefix v_myvar with the block label.

(*)

It cannot be done because the outer block's v_myvar is in scope but not visible at Line A.

Examine the following code. At Line A, we want to assign a value of 25 to the outer block's variable (V1). What must we do?

```
DECLARE
    v_myvar NUMBER; -- This is V1
BEGIN
    DECLARE
       v_myvar NUMBER := 8;
       BEGIN
          -- Line A
       END;
END;
```

Mark for Review
(1) Points

At Line A, code:
v_myvar := 25;

Label both blocks and at line A, code:
v_myvar := 25;

It cannot be done because the outer block's v_myvar is out of scope at Line A.

Label the outer block and (at Line A) dot-prefix v_myvar with the block label.

(*)

It cannot be done because the outer block's v_myvar is in scope but not visible at Line A.

. Examine the following code. What is the scope of variable v_myvar?

```
DECLARE
    v_myvar NUMBER;
BEGIN
    v_myvar := 6;
    DECLARE
```

```
      v_hervar NUMBER;
    BEGIN
      v_hervar := 4;
    END;
END;
```

Mark for Review
(1) Points

Only the outer block

Both the inner and the outer block (*)

Only the inner block

Neither block

Examine the following nested blocks. Line B causes an exception. What will be displayed when this code is executed?

```
DECLARE
    var_1 NUMBER;
BEGIN
    var_1 := 4;
    DECLARE
      var_2 NUMBER;
    BEGIN
      var_2 := 'Unhappy'; -- Line B
      var_1 := 8;
    END;
    var_1 := 12;
EXCEPTION
    WHEN OTHERS THEN
      DBMS_OUTPUT.PUT_LINE(var_1);
END;
```

Mark for Review
(1) Points

Unhappy

12

8

4 (*)

Examine the following code. Line A causes an exception. What will be displayed when the block is executed?

```
DECLARE
    x NUMBER := 10;
    y NUMBER;
BEGIN
    x := 15;
    y := 'Happy'; -- Line A
    x := 20;
EXCEPTION
```

```
    WHEN OTHERS THEN
    DBMS_OUTPUT.PUT_LINE(x);
END;
```

Mark for Review
(1) Points

10

20

15 (*)

Nothing is displayeExamine the following code. Line A causes an exception. What will be displayed when the block is executed?

```
DECLARE
    x NUMBER := 10;
    y NUMBER;
BEGIN
    x := 15;
    y := 'Happy'; -- Line A
    x := 20;
EXCEPTION
    WHEN OTHERS THEN
    DBMS_OUTPUT.PUT_LINE(x);
END;
```

Mark for Review
(1) Points

10

20

15 (*)

Nothing is displayed

What values will be displayed when the following code is executed?

```
DECLARE
    v_mynum NUMBER;
BEGIN
    v_mynum := 7;
    DECLARE
       v_mynum NUMBER;
    BEGIN
       DBMS_OUTPUT.PUT_LINE(v_mynum);
       v_mynum := 3;
    END;
    DBMS_OUTPUT.PUT_LINE(v_mynum);
END;
```

Mark for Review
(1) Points

3,3

3,7

Null, 7 (*)

Null, 3

What is wrong with the following statement?
DELETE from employees WHERE salary > (SELECT MAX(salary) FROM employees);
Mark for Review
(1) Points

You cannot code a subquery inside a DELETE statement.

You cannot use inequality operators such as "<" and ">" inside a DELETE statement.

Nothing is wrong, the statement will execute correctly. (*)

Is it possible to insert more than one row at a time using an INSERT statement with a VALUES clause? Mark for Review
(1) Points

No, you can only create one row at a time when using the VALUES clause. (*)

Yes, you can list as many rows as you want, just remember to separate the rows with commas.

No, there is no such thing as INSERT ... VALUES.

When inserting a row into a table, the VALUES clause must include a value for every column of the table. True or False? Mark for Review
(1) Points

True

False (*)

What is wrong with the following statement? MERGE INTO emps e USING new_emps ne ON (e.employee_id = ne.employee_id) WHEN MATCHED THEN UPDATE SET ne.salary = e.salary WHEN NOT MATCHED THEN INSERT VALUES (ne.employee_id, ne.first_name, ne.last_name, ... ne.salary, ...); Mark for Review
(1) Points

The UPDATE clause must include the target table name: UPDATE emps SET ....

The INSERT clause must include a column list as well as a list of column values.

The SET clause is trying to update the source table from the target table. (*)

Nothing is wrong, the statement will execute correctly.

Look at this SQL statement: MERGE INTO old_trans ot USING new_trans nt ON (ot.trans_id = nt.trans_id) .... ; OLD_TRANS is the source table and NEW_TRANS is the target table. True or false? Mark for Review
(1) Points

True

False (*)

To modify an existing row in a table, you can use the _____ statement. Mark for Review
(1) Points

MODIFY

INSERT

ALTER

UPDATE (*)

What would be the result of the following statement: DELETE employees; Mark for Review
(1) Points

Nothing, no data will be changed.

All rows in the employees table will be deleted. (*)

The statement will fail because it contains a syntax error.

The row with EMPOYEE_ID=100 will be deleted.

You want to modify existing rows in a table. Which of the following are NOT needed in your SQL statement? (Choose Two) Mark for Review
(1) Points

(Choose all correct answers)

A MODIFY clause (*)

An UPDATE clause

The name of the table

The name of the column(s) you want to modify.

A new value for the column you want to modify (this can be an expression or a subquery).

A WHERE clause. (*)

Does PL/SQL allow you to have a variable with the same name as a database column? Mark for Review
(1) Points

No

Yes (*)

What will happen when the following block is executed? DECLARE v_last employees.last_name%TYPE; v_first employees.first_name%TYPE; v_salary employees.salary%TYPE; BEGIN SELECT first_name, last_name INTO v_first, v_last, v_salary FROM employees WHERE employee_id=100; END; Mark for Review
(1) Points

The block will fail because the SELECT statement returns more than one row.

The block will fail because the SELECT is trying to read two columns into three PL/SQL variables. (*)

The block will fail because V_LAST was declared before V_FIRST.

The block will execute successfully, and the V_SALARY variable will be set to NULL.

When used in a PL/SQL block, which SQL statement must return exactly one row? Mark for Review
(1) Points

INSERT

UPDATE

SELECT (*)

MERGE

DELETE

PlSQL feedback midterm semister 1 part2

Look at this PL/SQL block: DECLARE v_count NUMBER; BEGIN SELECT COUNT(*) INTO v_count FROM employees WHERE salary > 50000; END; No employees earn more than $50000. Which of the following statements are true? (Choose two). Mark for Review
(1) Points

(Choose all correct answers)

The SELECT will return value 0 into V_COUNT. (*)

The SELECT will fail because it does NOT return exactly one row.

The block will fail because variable V_SALARY was not declared.

The SELECT returns exactly one row. (*)

The block will fail because no results are displayed to the user.

Which of the following is NOT a valid guideline for retrieving data in PL/SQL? Mark for Review
(1) Points

Terminate the SQL statement with a semicolon (;)

Do NOT use a WHERE clause in SELECT statements. (*)

Where possible, declare variables using the %TYPE attribute.

Specify the same number of variables in the INTO clause as database columns in the SELECT clause.

Incorrect Incorrect. Refer to Section 3.

It is good programming practice to create identifiers having the same name as column names. True or False? Mark for Review
(1) Points

True

False (*)

Which SQL statements can be used directly in a PL/SQL block? (Choose two.) Mark for Review
(1) Points

(Choose all correct answers)

GRANT EXECUTE ON ...

SELECT * INTO ... (*)

REVOKE SELECT ON ...

UPDATE employees SET... (*)

ALTER TABLE employees ...

Which one of these SQL statements can be directly included in a PL/SQL executable block? Mark for Review
(1) Points

IF... THEN...;

INSERT INTO...; (*)

SELECT * FROM DUAL;

SHOW USER;

There are three employees in department 90. What will be displayed when the following code is executed? DECLARE v_open CHAR(3) := 'NO'; BEGIN UPDATE employees SET job_id = 'ST_CLERK' WHERE department_id = 90; IF SQL%FOUND THEN v_open := 'YES'; END IF; DBMS_OUTPUT.PUT_LINE(v_open || ' ' || SQL%ROWCOUNT); END; Mark for Review
(1) Points

NO 3

YES 1

YES 3 (*)

Nothing will be displayed. The block will fail because you cannot use implicit cursor attributes directly in a call to DBMS_OUTPUT.PUT_LINE.

A PL/SQL block contains the following DML statement: UPDATE wf_countries SET population = population * 1.1 WHERE country_id = 229; Which kind of cursor is used for this statement? Mark for Review
(1) Points

An implicit cursor named "WF_COUNTRIES".

An implicit cursor named "SQL". (*)

An explicit cursor named "SQL".

An explicit cursor which must be declared and named by the PL/SQL programmer.

Employee_id 999 does not exist. What will happen when the following code is executed? DECLARE employee_id employees.employee_id%TYPE := 999; BEGIN UPDATE employees SET salary = salary * 1.1 WHERE employee_id = employee_id; END; Mark for Review
(1) Points

No rows are updated but the block completes successfully.

Every employee row is updated. (*)

An exception is raised because you cannot give a variable the same name as a table column.

An exception is raised because the UPDATE statement did not modify any rows.

You can use implicit cursor attributes such as SQL%ROWCOUNT directly inside a DML statement. For example: INSERT INTO log_table VALUES (SYSDATE, USER, SQL%ROWCOUNT); True or False? Mark for Review
(1) Points

True

False (*)

Which of the following SQL DML commands can be used inside a PL/SQL block? Mark for Review
(1) Points

INSERT and UPDATE only.

UPDATE and DELETE only.

INSERT, UPDATE and DELETE only.

INSERT, UPDATE, DELETE and MERGE. (*)

Which of the following use an implicit cursor? Mark for Review
(1) Points

DML statements only.

SELECT statements only.

DML statements and SELECT statements which return a single row. (*)

COMMIT and ROLLBACK statements only.

Examine the following code: BEGIN
INSERT INTO animals VALUES ('aa','aardvarks');
SAVEPOINT sp_1;
INSERT INTO animals VALUES ('bb','big birds');
SAVEPOINT sp_2;
ROLLBACK TO sp_1;
INSERT INTO animals VALUES ('cc','cool cats');
COMMIT;
END;
Which row(s) will be in the ANIMALS table after this block is executed? Mark for Review
(1) Points

cool cats

big birds and cool cats

aardvaarks and cool cats (*)

aardvaarks, big birds and cool cats

How many INSERTs can you have in one transaction? Mark for Review
(1) Points

One

As many as you want until you do a COMMIT or ROLLBACK. (*)

As many as you can execute before the database does an AUTOSAVE.

As many as you want until a different DML statement (UPDATE, DELETE or MERGE) is executed.

In a PL/SQL block, where can you code a COMMIT statement? Mark for Review
(1) Points

In any section of the block: Declaration, Executable, or Exception.

Only the Executable section.

In the Executable and/or the Exception sections. (*)

Nowhere; the COMMIT statement must be outside the block.

How many transactions are in the following block?

```
BEGIN
    INSERT INTO countries (country_id, country_name)
       VALUES ('XA', 'Xanadu');
    INSERT INTO countries (country_id, country_name)
       VALUES ('NV', 'Neverland');
    UPDATE countries SET country_name='Deutchland'
       WHERE country_id='DE';
    UPDATE countries SET region_id=1
       WHERE country_name LIKE '%stan';
END;
```

How many transactions are shown above?
Mark for Review
(1) Points

Four; each DML is a separate transaction

Two; both the INSERTs are one transaction and both the UPDATEs are a second transaction.

It depends on how many rows are updated - there will be a separate transaction for each row.

One (*)

We want to execute one of three statements depending on whether the value in V_VAR is 10, 20 or some other value. What should be coded at Line A? IF v_var = 10 THEN statement1; -- Line A statement2; ELSE statement3; END IF; Mark for Review
(1) Points

ELSE IF v_var = 20 THEN

ELSIF v_var = 20

ELSIF v_var = 20 THEN (*)

IF v_var = 20 THEN

What is wrong with the following trivial IF statement:

```
IF (v_job='President')
THEN v_salary := 10000;
```

Mark for Review
(1) Points

IF and THEN must be on the same line: IF (v_job='President') THEN ...

The condition should be coded: IF (v_job := 'President')

END IF; is missing (*)

ELSE is missing

You want to repeat a set of statements 100 times, incrementing a counter each time. What kind of PL/SQL control structure would you use? Mark for Review
(1) Points

IF...THEN...ELSE

IF...THEN...ELSIF...ELSE

CASE...WHEN...THEN

A loop. (*)

Look at the following (badly written) code:

```
age := 5; IF age<30 THEN mature := 'adult';
ELSIF age<22 THEN mature := 'teenager';
ELSIF age<13 THEN mature := 'child';
END IF;
DBMS_OUTPUT.PUT_LINE(mature);
```

What will be displayed when this code is executed?
Mark for Review
(1) Points

child

teenager

adult (*)

adultteenagerchi

Which one of the following is correct syntax for an IF statement? Mark for Review
(1) Points

IF condition THEN DO statement1; statement2; END IF;

IF condition THEN statement1; statement2; END IF; (*)

IF condition THEN statement1; statement2; ENDIF;

IF condition THEN statement1; AND statement2; END IF;

What will be displayed when this block is executed? DECLARE v_bool1 BOOLEAN := NULL; v_bool2 BOOLEAN := NULL; v_char VARCHAR(10) := 'Start'; BEGIN IF (v_bool1 = v_bool2) THEN v_char:='Equal'; ELSE v_char:='Not equal'; END IF; DBMS_OUTPUT.PUT_LINE(v_char); END; Mark for Review
(1) Points

Equal

Not equal (*)

Start

Nothing will be displayed. The block will fail because you cannot compare two null values.

Which of the following statements are true about PL/SQL conditional control structures such as IF ... , CASE ... and loops? Mark for Review
(1) Points

They allow the programmer to use logical tests to determine which statements are executed and which are not.

They allow a set of statements to be executed repeatedly (i.e. more than once).

They determine a course of action based on conditions.

All of the above. (*)

What will be displayed when this block is executed? DECLARE v_bool1 BOOLEAN := TRUE; v_bool2 BOOLEAN; v_char VARCHAR(4) := 'up'; BEGIN IF (v_bool1 AND v_bool2) THEN v_char:='down'; ELSE v_char:='left'; END IF; DBMS_OUTPUT.PUT_LINE(v_char); END; Mark for Review
(1) Points

up

down

left (*)

null

Look at the following code:

```
DECLARE
x BOOLEAN := FALSE;
y BOOLEAN := FALSE;
z BOOLEAN ;
BEGIN
```

```
z := (x OR NOT y);
-- Line A
....
END;
```

What is the value of Z at Line A?

Mark for Review

(1) Points

True (*)

False

NULL

An error will occur because you cannot combine two Boolean variables using "NOT".

Examine the following code:

```
DECLARE
v_score NUMBER(3);
v_grade CHAR(1);
BEGIN
v_grade := CASE v_score
-- Line A
....
```

The CASE expression must convert a numeric score to a letter grade: 90 -> A, 80 -> B, 70 -> C and so on. What should be coded at Line A?

Mark for Review

(1) Points

WHEN 90 THEN grade := 'A'

WHEN 90 THEN v_grade := 'A';

WHEN 90 THEN 'A' (*)

WHEN 90 THEN 'A';

What will be displayed when the following block is executed?

```
DECLARE
v_age1 NUMBER(3);
v_age2 NUMBER(3);
v_message VARCHAR2(20);
BEGIN
CASE
WHEN v_age1 = v_age2 THEN v_message := 'Equal';
WHEN v_age1 <> v_age2 THEN v_message := 'Unequal';
ELSE v_message := 'Undefined';
END CASE;
```

```
DBMS_OUTPUT.PUT_LINE(v_message);
END;
```

Mark for Review
(1) Points

Equal

Undefined (*)

Unequal

Nothing will be displayed because V_MESSAGE is set to NULL.

Incorrect Incorrect. Refer to Section 4.

How must you end a CASE expression? Mark for Review
(1) Points

END; (*)

ENDIF;

END CASE;

ENDCASE;

Examine the following code:

```
DECLARE
v_score NUMBER(3);
v_grade CHAR(1);
BEGIN
CASE v_score
-- Line A
....
```

The CASE statement must convert a numeric score to a letter grade: 90 -> A, 80 -> B, 70 -> C and so on.

What should be coded at Line A?
Mark for Review
(1) Points

WHEN 90 THEN v_grade := 'A'

WHEN 90 THEN v_grade := 'A'; (*)

WHEN 90 THEN 'A'

```
        WHEN 90 THEN 'A';
```

What will be displayed when the following block is executed?

```
DECLARE
v_age NUMBER(3);
v_gender VARCHAR2(6) := 'Female';
v_status VARCHAR2(20);
BEGIN
CASE
WHEN v_age >= 18 AND v_gender = 'Male' THEN v_status := 'Adult Male';
WHEN v_age >= 18 AND v_gender = 'Female' THEN v_status := 'Adult Female';
WHEN v_age < 18 AND v_gender = 'Male' THEN v_status := 'Junior Male';
WHEN v_age < 18 AND v_gender = 'Female' THEN v_status := 'Junior Female';
ELSE v_status := 'Other Value';
END CASE;
DBMS_OUTPUT.PUT_LINE(v_status);
END;
```

Mark for Review
(1) Points

Adult Male

Junior Female

Other Value (*)

Nothing will be displayed because V_STATUS is set to NULL.

Examine the following code:

```
DECLARE
v_a BOOLEAN;
v_b BOOLEAN := FALSE;
v_c BOOLEAN ;
BEGIN
v_c := (v_a AND v_b);
-- Line A
....
END;
```

What is the value of V_C at Line A?
Mark for Review
(1) Points

True

False (*)

NULL

Undefined

How must you end a CASE statement? Mark for Review
(1) Points

END;

END CASE; (*)

END IF;

ENDCASE;

Which kind of loop is this?

```
i := 10;
LOOP
    i := i + 1;
    EXIT WHEN i > 30;
END LOOP;
```

Mark for Review
(1) Points

A FOR loop.

A WHILE loop.

A basic loop. (*)

An infinite loop.

A nested loop.

What are the three kinds of loops in PL/SQL? Mark for Review
(1) Points

ascending, descending, unordered

infinite, finite, recursive

IF, CASE, LOOP

FOR, WHILE, basic (*

PlSQL feedback midterm semister 1 part2

How many EXIT statements can be coded inside a basic loop? Mark for Review
(1) Points

None.

One only.

Two.

As many as you need, there is no limit. (*)

ou want to calculate and display the multiplication table for "sevens": 7x1=7, 7x2=14, 7x3=21 and so on. Which kind of PL/SQL construct is best for this? Mark for Review
(1) Points

A loop (*)

A CASE statement

IF ... END IF;

A Boolean variable

For which one of these tasks should you use a PL/SQL loop? Mark for Review
(1) Points

Updating the salary of one employee.

Executing the same set of statements repeatedly until a condition becomes true. (*)

Deciding whether a value is within a range of numbers.

Making a decision based on whether a condition is true or not.

What will be displayed when this block is executed?

```
DECLARE
v_count NUMBER := 10;
v_result NUMBER;
BEGIN
LOOP
v_count := v_count - 1;
EXIT WHEN v_count < 5;
v_result := v_count * 2;
END LOOP;
DBMS_OUTPUT.PUT_LINE(v_result);
END;
```

Mark for Review
(1) Points

8

10 (*)

12

NULL

Examine the following code:

```
DECLARE
v_count NUMBER := 0;
v_string VARCHAR2(20);
BEGIN
LOOP
v_string := v_string || 'x';
IF LENGTH(v_string) > 10 THEN
EXIT;
END IF;
v_count := v_count + 1;
END LOOP;
DBMS_OUTPUT.PUT_LINE(v_count);
END;
```

What will be displayed when this block is executed?

Mark for Review
(1) Points

9

10 (*)

11

xxxxxxxxx

Look at this code:

```
DECLARE
v_bool BOOLEAN := TRUE;
v_date DATE;
BEGIN
LOOP
EXIT WHEN v_bool;
SELECT SYSDATE INTO v_date FROM dual;
END LOOP;
END;
```

How many times will the SELECT statement execute?

Mark for Review
(1) Points

Once.

Twice.

Never (the SELECT will not execute at all) (*)

An infinite number of times because the EXIT condition will never be true

You should use a WHILE loop when the number of iterations of the loop is known in advance. True or False? Mark for Review
(1) Points

True

False (*)

Look at the following block:

```
DECLARE
v_date DATE := SYSDATE;
BEGIN
WHILE v_date < LAST_DAY(v_date) LOOP
v_date := v_date + 1;
END LOOP;
DBMS_OUTPUT.PUT_LINE(v_date);
END;
```

If today's date is 17th April 2007, what will be displayed when this block executes? Mark for Review
(1) Points

01-MAY-07

31-DEC-07

4/30/2007 (*)

4/17/2007

In a FOR loop, an explicitly declared counter is automatically incremented by 1 for each iteration of the loop. True or False? Mark for Review
(1) Points

True

False (*)

Which statement best describes when a FOR loop should be used? Mark for Review
(1) Points

When an EXIT WHEN statement must be coded.

When an implicitly declared counter must increase by 1 in each iteration of the loop. (*)

When we want to exit from the loop when a Boolean variable becomes FALSE.

When the statements inside the loop must execute at least once.

You want a loop that counts backwards from 10 through 1. How do you code that? Mark for Review
(1) Points

FOR i IN 10 .. 1 LOOP

FOR i IN 1 .. 10 BY -1 LOOP

FOR i IN REVERSE 1 .. 10 LOOP (*)

FOR i IN REVERSE 10 .. 1 LOOP

Look at this code fragment:

```
FOR i IN 1 .. 3 LOOP
i := 4;
DBMS_OUTPUT.PUT_LINE('The counter is: ' || i);
END LOOP;
```

How many lines of output will be displayed?
Mark for Review
(1) Points

One

Three

Four

The block will fail because you cannot change the value of i inside the loop. (*)

In a WHILE loop, the controlling condition is checked at the start of each iteration. True or False? Mark for Review
(1) Points

True (*)

False

Look at the following code fragment:

```
i := 2;
WHILE i < 3 LOOP
i := 4;
DBMS_OUTPUT.PUT_LINE('The counter is: ' || i);
END LOOP;
```

How many lines of output will be displayed?
Mark for Review
(1) Points

No lines

One line (*)

Two lines

The block will fail because you cannot use DBMS_OUTPUT.PUT_LINE inside a loop.

Look at the following code:

```
DECLARE
v_blue NUMBER(3) := 0;
v_red NUMBER(3) := 0;
BEGIN
<<blue>> LOOP
v_blue := v_blue + 1;
EXIT WHEN v_blue > 10;
<<red>> LOOP
v_red := v_red + 1;
EXIT WHEN v_red > 10;
-- Line A
END LOOP red;
END LOOP blue;
END;
```

What should you code at Line A to exit from the outer loop?
Mark for Review
(1) Points

EXIT;

EXIT red;

EXIT <<blue>>;

EXIT blue; (*)

PlSQL feedback midterm semister 1 part2

What will be displayed when the following block is executed?:

```
DECLARE
x NUMBER(6) := 0 ;
BEGIN
FOR i IN 1..10 LOOP
FOR j IN 1..5 LOOP
x := x+1 ;
END LOOP;
END LOOP;
DBMS_OUTPUT.PUT_LINE(x);
END;
```

Mark for Review
(1) Points

5

10

15

50 (*)

Which one of these statements about using nested loops is true? Mark for Review
(1) Points

All the loops must be labelled

The outer loop must be labelled, but the inner loop need not be labelled

The outer loop must be labelled if you want to exit the outer loop from within the inner loop (*)

Both loops can have the same label

When the following code is executed, how many lines of output will be displayed?

```
BEGIN
FOR i IN 1..5 LOOP
FOR j IN 1..8 LOOP
DBMS_OUTPUT.PUT_LINE(i || ',' || j);
END LOOP;
DBMS_OUTPUT.PUT_LINE(i);
END LOOP;
END;
```

Mark for Review
(1) Points

80

45 (*)

14

41

You cannot OPEN or CLOSE an implicit cursor. Why not? Mark for Review (1) Points

Because an implicit cursor is always called SQL.

Because an implicit cursor is OPENed and CLOSEd automatically by Oracle. (*)

When must you declare and use an explicit cursor? Mark for Review (1) Points

You need to UPDATE more than one row in a table.

You want to use a MERGE statement.

You need to SELECT more than one row from a table. (*)

You want to be able to ROLLBACK a transaction if needed

One (and only one) employee has LAST_NAME = 'Grant'. You need to code:

SELECT ... FROM employees WHERE last_name = 'Grant';

Which type of cursor should you use, and why?

Mark for Review (1) Points

An implicit cursor, because there is only one 'Grant'.

An implicit cursor, because SELECT is a SQL statement and implicit cursors are always called "SQL".

An explicit cursor, because there could be more than one 'Grant' in the future. (*)

An explicit cursor, because you can use an implicit cursor only for DML statements.

There are 8 countries in REGION_ID 13 (Central America). What will happen when the following code is executed?

DECLARE

```
CURSOR country_curs IS SELECT country_name FROM wf_countries
WHERE region_id = 13;
v_country_name wf_countries.country_name%TYPE;
BEGIN
OPEN country_curs;
WHILE country_curs%FOUND
LOOP
FETCH country_curs INTO v_country_name;
DBMS_OUTPUT.PUT_LINE(v_country_name);
END LOOP;
CLOSE country_curs;
END;
```

Mark for Review
(1) Points

Eight rows will be fetched and displayed successfully.

The last seven rows will be fetched and displayed.

The block will execute, but no rows will be displayed. (*)

The block will fail because you can not use a WHILE loop with an explicit cursor.

None of the above.

Examine the following code:

```
DECLARE
CURSOR dept_curs IS SELECT department_name FROM departments;
v_dept_name departments.department_name%TYPE;
BEGIN
OPEN dept_curs;
LOOP
FETCH dept_curs INTO v_dept_name;
DBMS_OUTPUT.PUT_LINE(v_dept_name);
EXIT WHEN dept_curs%NOTFOUND;
END LOOP;
CLOSE dept_curs;
END;
```

There are 10 rows in the DEPARTMENTS table. What will happen when this code is executed?

Mark for Review
(1) Points

10 rows will be displayed.

10 rows will be displayed, followed by a row of NULL values.

The last row will be displayed twice. (*)

A NO_DATA_FOUND exception will be raised.

The loop will execute for ever; the same 10 rows will be displayed over and over again.

Which one of the following statements is NOT true? Mark for Review
(1) Points

You can use ORDER BY when declaring an explicit cursor.

You can not use an INTO clause when declaring an explicit cursor.

An explicit cursor can select from only one table. No joins are allowed. (*)

An explicit cursor must be DECLAREd before it can be OPENed.

What is wrong with the following code?

```
DECLARE
CURSOR emp_curs IS SELECT last_name, salary FROM employees;
v_last_name employees.last_name%TYPE;
v_salary employees.salary%TYPE;
BEGIN
FETCH emp_curs INTO v_last_name, v_salary;
OPEN emp_curs;
FETCH emp_curs INTO v_last_name, v_salary;
CLOSE emp_curs;
END;
```

Mark for Review
(1) Points

When FETCHing more than one row, you MUST use a loop.

The cursor declaration does not include a WHERE condition.

The cursor declaration does not include an INTO clause.

The first row is FETCHed before the cursor is OPENed. (*)

You have declared a cursor EMP_CURSOR to select many rows from the EMPLOYEES table. The following five statements will be in the executable section:

```
A. FETCH emp_cursor INTO v_empno,v_last_name;
B. OPEN emp_cursor;
C. END LOOP;
D. CLOSE emp_cursor;
E. LOOP
```

In which order should you code these statements?
Mark for Review

(1) Points

B, E, A, C, D (*)

E, B, A, C, D

B, E, A, D, C

B, A, E, D, C

You execute the following code:

```
DECLARE
CURSOR emp_curs IS SELECT last_name FROM employees;
v_last_name employees.last_name%TYPE;
BEGIN
OPEN emp_curs;
LOOP -- Point A
FETCH emp_curs INTO v_last_name;
EXIT WHEN emp_curs%NOTFOUND;
DBMS_OUTPUT.PUT_LINE(v_last_name);
END LOOP;
CLOSE emp_curs;
END;
```

At Point A (after you have OPENed the cursor) another user updates an employee's last_name from 'Smith' to 'Jones' and immediately COMMITs.

When your block FETCHes this row, which value will be fetched and displayed?
Mark for Review

(1) Points

1

Smith (*)

Jones

Smith and Jones (the row will be fetched twice)

An INVALID_CURSOR exception will be raised when you try to FETCH the row.

Which of the following best describes the difference between implicit and explicit cursors? Mark for Review

(1) Points

Implicit cursors are used for SELECT statements, while explicit cursors are used for DML statements.

Implicit cursor are named by the PL/SQL programmer, while explicit cursors are always named SQL.

Implicit cursors are defined automatically by Oracle, while explicit cursors must be declared by the PL/SQL programmer. (*)

Implicit cursors store rows on disk, while explicit cursors store rows in memory.

Which one of the following explicit cursor declarations is NOT valid? Mark for Review
(1) Points

CURSOR country_curs IS
SELECT country_name, region_name
FROM wf_countries c, wf_world_regions r
WHERE c.region_id = r.region_id;

CURSOR country_curs IS
SELECT country_name INTO v_country_name
FROM wf_countries;

(*)

CURSOR country_curs IS
SELECT country_name
FROM wf_countries
ORDER BY population DESC;

CURSOR country_curs IS
SELECT country_name
FROM wf_countries
WHERE region_id IN
(SELECT region_id FROM wf_world_regions
WHERE LOWER(region_name) LIKE '%asia%');

What is wrong with the following code?

```
DECLARE
CURSOR dept_curs IS SELECT department_name FROM departments;
v_dept_name departments.department_name%TYPE;
BEGIN
OPEN dept_curs;
LOOP
FETCH dept_curs INTO v_dept_name;
EXIT WHEN dept_curs%NOTFOUND;
DBMS_OUTPUT.PUT_LINE(v_dept_name);
CLOSE dept_curs;
END LOOP;
END;
```

Mark for Review

(1) Points

Nothing is wrong, all the rows will be FETCHed and displayed.

The OPEN statement should be inside the loop.

The EXIT WHEN ... statement should be coded outside the loop.

The CLOSE statement should be coded after END LOOP; (*)

The loop should be a WHILE loop, not a basic loop.

Examine the following code:

```
DECLARE
CURSOR country_curs IS
SELECT country_id, country_name
FROM wf_countries
ORDER BY country_name;
v_country country_curs%ROWTYPE;
BEGIN
OPEN country_curs;
LOOP
FETCH country_curs INTO v_country;
EXIT WHEN country_curs%NOTFOUND;
------- Line A
END LOOP;
CLOSE country_curs;
END;
```

You want to display the id and name of each FETCHed country. What would you code at Line A?

Mark for Review

(1) Points

DBMS_OUTPUT.PUT_LINE(country_id || ' ' || country_name);

DBMS_OUTPUT.PUT_LINE(v_country(country_id) || ' ' || v_country(country_name));

DBMS_OUTPUT.PUT_LINE(country_curs.country_id || ' ' || country_curs.country_name);

DBMS_OUTPUT.PUT_LINE(v_country.country_id || ' ' || v_country.country_name); (*)

You can reference explicit cursor attributes directly in a SQL statement. True or False? Mark for Review

(1) Points

True

False (*)

Look at the following code:

```
DECLARE
CURSOR emp_cursor IS
SELECT employee_id, last_name, salary FROM employees;
v_empcurs emp_cursor%ROWTYPE;
```

What is the data type of V_EMPCURS?
Mark for Review
(1) Points

Scalar

Record (*)

Cursor

Row

You have declared the following cursor:

```
CURSOR country_curs IS
SELECT * FROM wf_countries
ORDER BY country_name;
```

There are over 200 rows in the WF_COUNTRIES table, but you want to fetch and display only the first 25 rows.

How would you exit from the FETCH loop?
Mark for Review
(1) Points

EXIT WHEN country_curs%FOUND(25);

EXIT WHEN country_curs%ROWCOUNT > 25; (*)

EXIT WHEN ROWCOUNT > 25;

WHEN country_curs > 25 THEN EXIT; END IF;

Look at these declarations:

```
DECLARE
CURSOR dept_loc_cursor IS
```

```
SELECT department_id, department_name, location_name
FROM departments d, locations l
WHERE d.location_id = l.location_id;
v_dept_loc dept_loc_cursor%ROWTYPE;
```

How many fields does V_DEPT_LOC contain?
Mark for Review
(1) Points

Two, because the cursor joins two tables

Four

Three (*)

None

Which of the following explicit cursor attributes evaluates to TRUE if the most recent FETCH returns a row? Mark for Review
(1) Points

%ISOPEN

%NOTFOUND

%FOUND (*)

%ROWCOUNT

How must you reference one field which is part of a PL/SQL record? Mark for Review
(1) Points

field_name.record_name

record_name.field_name (*)

record_name(field_name)

field_name OF record_name

It cannot be done.

You have declared a cursor as follows:

```
CURSOR loc_curs IS SELECT * FROM locations;
```

How should you code a FOR loop to use this cursor?
Mark for Review
(1) Points

FOR loc_rec IN 1 .. loc_curs%ROWCOUNT LOOP ...

WHILE loc_rec IN loc_curs LOOP ...

FOR loc_curs IN loc_rec LOOP ...

IF loc_rec IN loc_curs LOOP ...

FOR loc_rec IN loc_curs LOOP ... (*)

Which of the following is a benefit of using a cursor FOR loop? Mark for Review
(1) Points

The exception handling is done automatically. .

The OPEN, CLOSE, FETCH and EXIT from the loop are done automatically. (*)

You can OPEN the same cursor twice at the same time.

Because there is less code, the loop executes faster.

%ROWCOUNT increments automatically each time a row is FETCHed.

What is wrong with the following piece of code?

```
BEGIN
FOR emp_record IN emp_cursor LOOP
DBMS_OUTPUT.PUT_LINE(emp_record.last_name);
END LOOP;
IF emp_record.last_name = 'Patel' THEN ...
```

Mark for Review
(1) Points

EMP_RECORD has not been explicitly declared.

The cursor has not been OPENed.

You cannot reference EMP_RECORD outside the loop. (*)

It should read: DBMS_OUTPUT.PUT_LINE(emp_cursor.last_name);

Nothing is wrong, the code will execute correctly.

What is the DISadvantage of using a cursor FOR loop with a subquery? Mark for Review
(1) Points

You cannot reference cursor attributes such as %NOTFOUND. (*)

The execution speed is slower.

You cannot declare the cursor in the declaration section.

You cannot use the cursor to join two or more tables.

There are no disadvantages.

Look at the following code:

```
DECLARE
CURSOR emp_cursor IS SELECT * FROM employees;
BEGIN
FOR emp_record IN emp_cursor LOOP
DBMS_OUTPUT.PUT_LINE( --Point A -- );
END LOOP;
END;
```

To display the salary of an employee, what code should you write at Point A?
Mark for Review
(1) Points

emp_record.salary (*)

emp_cursor.salary

employees.salary

emp_record.employees.salary

TO_CHAR(salary)

Which one of the following is a valid cursor FOR loop with a subquery? Mark for Review
(1) Points

FOR emp_rec IN (SELECT last_name || first_name FROM employees) LOOP ...

FOR emp_rec IN (SELECT UPPER(last_name) FROM employees) LOOP ...

FOR emp_rec IN SELECT last_name, salary*12 "ANNSAL" FROM employees LOOP ...

FOR emp_rec IN (SELECT last_name, salary*12 "ANNSAL" FROM employees) LOOP ... (*)

None of the above.

The following cursor has been declared:

```
CURSOR emp_curs
(p_dept_id employees.department_id%TYPE,
p_job_id employees.job_id%TYPE) IS
SELECT * FROM employees
WHERE department_id = p_dept_id
AND job_id = p_job_id;
```

Which of the following will correctly open the cursor?
Mark for Review
(1) Points

OPEN emp_curs(20);

FOR emp_rec IN emp_curs(20) LOOP ...

OPEN emp_curs('IT_PROG', 20);

FOR emp_rec IN emp_curs(20,'IT_PROG') LOOP ... (*)

FOR emp_rec IN emp_curs(p_dept_id p_job_id) LOOP .

You want to use explicit cursors to fetch and display all the countries in a specific region. There are 19 rows in the WF_WORLD_REGIONS table. You want to use a different region each time the cursor is opened. How many cursors should you declare? Mark for Review
(1) Points

19 cursors, all in the same PL/SQL block.

19 cursors in 19 PL/SQL blocks (one in each block).

20 cursors, in case an extra row is inserted into WF_WORLD_REGIONS later.

One cursor with a parameter in the WHERE clause. (*)

None of the above

Look at the following code:

```
DECLARE
CURSOR emp_curs (p_dept_id employees.department_id%TYPE) IS
SELECT * FROM employees
WHERE department_id = p_dept_id;
v_emp_rec emp_curs%ROWTYPE;
v_deptid NUMBER(4) := 50;
BEGIN
OPEN emp_curs( -- Point A --);
....
```

You want to open the cursor, passing value 50 to the parameter. Which of the following are correct at Point A?
Mark for Review
(1) Points

50

v_deptid

100 / 2

All of the above. (*)

Using parameters with a cursor, you can open and close the cursor several times in a block, returning a different active set each time. True or False? Mark for Review
(1) Points

True (*)

False

What is wrong with the following cursor declaration?

```
CURSOR dept_curs (p_loc_id NUMBER(4)) IS
SELECT * FROM departments
WHERE location_id = p_loc_id;
```

Mark for Review
(1) Points

You cannot reference a cursor parameter in a WHERE clause.

The parameter should be coded as: (p_loc_id NUMBER) (*)

The parameter should be coded as: (p_loc_id IN NUMBER)

Nothing is wrong, the cursor declaration is correct.

What is wrong with the following cursor declaration?

```
CURSOR dept_curs (p_loc_id NUMBER(4)) IS
SELECT * FROM departments
WHERE location_id = p_loc_id;
```

Mark for Review
(1) Points

You cannot reference a cursor parameter in a WHERE clause.

The parameter should be coded as: (p_loc_id NUMBER) (*)

The parameter should be coded as: (p_loc_id IN NUMBER)

Nothing is wrong, the cursor declaration is correct.

You declare a cursor as a join of two tables:

```
CURSOR emp_dept_curs IS
SELECT last_name, salary, department_name
FROM employees e, departments d
WHERE e.department_id = d.department_id
-- Point A -- ;
```

You want to lock fetched rows from EMPLOYEES, but NOT lock fetched rows from DEPARTMENTS.

Which of the following is correct at Point A?
Mark for Review
(1) Points

FOR UPDATE

FOR UPDATE of salary (*)

FOR UPDATE OF employees

FOR UPDATE (last_name)

Why can we NOT code:

```
INSERT INTO table-name
WHERE CURRENT OF cursor_name;
```

Mark for Review
(1) Points

Because the syntax is wrong. An INSERT statement must have a VALUES (....) clause.

Because the syntax is wrong. It should be: INSERT INTO cursor-name .... WHERE CURRENT OF table-name;

Because WHERE CURRENT OF ... modifies the most recently FETCHed row, and you cannot FETCH a row that is not in the table yet. (*)

Because another user has locked the rows and not committed.

Nothing is wrong; we CAN code: INSERT .... WHERE CURRENT OF ... ;

You have declared a cursor as SELECT .... FOR UPDATE; You have OPENed the cursor and

locked the FETCHed rows. When are these row locks released? Mark for Review
(1) Points

When an UPDATE ... WHERE CURRENT OF cursor_name; is executed.

When you CLOSE the cursor.

When your block finishes executing.

Using parameters with a cursor, you can open and close the cursor several times in a block, returning a different active set each time. True or False? Mark for Review
(1) Points

True (*)

False

When you explicitly COMMIT or ROLLBACK your transaction. (*)

When another user tries to SELECT the rows.

You want to fetch rows from the EMPLOYEES table. You want to lock the fetched rows, to prevent other users from updating them.
You declare the following cursor:

```
CURSOR emp_curs IS
SELECT employee_id, last_name, salary
FROM employees
-- Line A -- ;
```

What should you code at Line A?
Mark for Review
(1) Points

FOR LOCK

FOR UPDATE OF employees

FOR UPDATE (*)

FOR UPDATE (employees)

You have declared the following cursor:

```
CURSOR country_curs IS
SELECT country_id, country_name
FROM wf_countries
FOR UPDATE WAIT 10;
```

Another user updates a row in WF_COUNTRIES but does not COMMIT the update. What will

happen when you OPEN country_curs; ?
Mark for Review
(1) Points

A LOCKED_ROWS exception is raised immediately.

The other user's transaction is automatically rolled back.

Your session waits indefinitely until the other user COMMITs.

Your session waits for 10 seconds, and then returns control to your block so that it can continue to execute. (*)

Your block fails because you should have coded: FOR UPDATE WAIT (10);

What is the difference between the following two blocks of code?

```
--Block A
DECLARE
    CURSOR emp_cursor IS
    SELECT employee_id, last_name
    FROM employees
    WHERE department_id = 80
    FOR UPDATE OF salary;

--Block B
DECLARE
    CURSOR emp_cursor IS
    SELECT employee_id, last_name
    FROM employees
    WHERE department_id = 80
    FOR UPDATE OF salary
    NOWAIT;
```

Mark for Review
(1) Points

There is no difference; the programs behave exactly the same way.

In Block A, the program waits indefinitely until the rows are available. In Block B, the program returns control immediately so that it can do other work. (*)

In Block A, the program waits indefinitely until the rows are available. In Block B, control is returned to your program after 5 seconds so that it can do other work.

When can we use the WHERE CURRENT OF clause? Mark for Review
(1) Points

Only with an UPDATE, not with a DELETE.

Only with a DELETE, not with an UPDATE.

When the cursor is declared as SELECT ... FOR UPDATE ...; (*)

When the cursor is based on a single table (not on a join).

When the cursor has not been OPENed.

Assume that table BIGDEPTS contains 100 rows, and table BIGEMPS contains 1000 rows, with 10 employees in each department. Consider the following code:

```
DECLARE
CURSOR bigdept_cur IS
SELECT * FROM bigdepts;
CURSOR bigemp_cur IS
SELECT * FROM bigemps;
BEGIN
FOR dept_rec IN bigdept_cur LOOP
DBMS_OUTPUT.PUT_LINE
(dept_rec.department_name);
FOR emp_rec IN bigemp_cur LOOP
IF emp_rec.department_id=dept_rec.department_id
THEN DBMS_OUTPUT.PUT_LINE
(emp_rec.last_name);
END IF;
END LOOP;
END LOOP;
END;
```

Why is this code inefficient?
Mark for Review
(1) Points

It locks both tables unnecessarily.

It is using two cursors when one cursor is enough.

It is doing a Cartesian Product, joining every employee with every department and displaying 1100 lines of output.

It reads 1000 employee rows every time BIGEMP_CUR is OPENed, and then ignores 990 of them. (*)

It is using cursor FOR loops, which are less efficient than OPENing and CLOSEing the cursors

Which of the following is a good reason to use two cursors in a single PL/SQL block?
Mark for Review
(1) Points

To allow one cursor to be opened twice at the same time.

When two tables are related to each other (often by a foreign key) and we want to produce a multilevel report using data from both tables. (*)

To allow rows to be locked as they are FETCHed.

To speed up the execution of the PL/SQL block.

It is the only way to declare a cursor with a parameter.

Which of the following is NOT allowed when using multiple cursors with parameters? Mark for Review
(1) Points

You cannot use cursor FOR loops.

You cannot declare the cursors FOR UPDATE.

You cannot declare a cursor based on a join.

You cannot OPEN more than one cursor at the same time.

None of the above, they are all allowed. (*)

You want to produce a report which displays each department and (immediately after each department) a list of employees who work in that department. You declare a DEPARTMENTS cursor as:

```
CURSOR dept_curs IS
SELECT * FROM departments
ORDER BY department_id;
```

How could you declare the EMPLOYEES cursor? (Choose two).

Mark for Review
(1) Points

(Choose all correct answers)

CURSOR emp_curs IS SELECT * FROM employees;

CURSOR emp_curs (p_dept_id NUMBER) IS SELECT * FROM employees WHERE department_id = p_dept_id; (*)

CURSOR emp_curs IS SELECT * FROM employees ORDER BY department_id;

CURSOR emp_curs (p_dept_id departments.department_id%TYPE) IS SELECT * FROM employees WHERE department_id = p_dept_id; (*)

CURSOR emp_curs IS SELECT * FROM employees WHERE department_id =

departments.department_id;

Examine the following code:

```
DECLARE
CURSOR region_cur IS
SELECT * FROM wf_world_regions;
v_region_rec region_cur%ROWTYPE;
CURSOR country_cur (p_region_id NUMBER) IS
SELECT * FROM wf_countries
WHERE region_id = p_region_id;
v_country_rec country_cur%ROWTYPE;
BEGIN
OPEN region_cur;
LOOP
FETCH region_cur INTO v_region_rec;
EXIT WHEN region_cur%NOTFOUND;
DBMS_OUTPUT.PUT_LINE
(v_region_rec.region_name);
-- Line A --
LOOP
FETCH country_cur INTO v_country_rec;
EXIT WHEN country_cur%NOTFOUND;
.......
```

What would you code at Line A?
Mark for Review
(1) Points

OPEN country_cur (p_region_id);

OPEN country_cur (wf_world_regions.region_id);

OPEN country_cur (v_region_rec.region_id); (*)

OPEN country_cur (region_cur.region_id);

OPEN country_cur;

Assume your schema contains 25 tables. How many explicit cursors can you declare and use within a single PL/SQL block? Mark for Review
(1) Points

Only one.

As many as you need - there is no limit. (*)

A maximum of three.

As many as you need, but only one of them can be open at any time.

A maximum of 25 (one for each table in your schema).

Assume your schema contains 25 tables. How many explicit cursors can you declare and use within a single PL/SQL block? Mark for Review
(1) Points

Only one.

As many as you need - there is no limit. (*)

A maximum of three.

As many as you need, but only one of them can be open at any time.

A maximum of 25 (one for each table in your schema).

ou have declared a cursor as SELECT .... FOR UPDATE; You have OPENed the cursor and locked the FETCHed rows. When are these row locks released? Mark for Review
(1) Points

When an UPDATE ... WHERE CURRENT OF cursor_name; is executed.

When you CLOSE the cursor.

When your block finishes executing.

When you explicitly COMMIT or ROLLBACK your transaction. (*)

When another user tries to SELECT the rows.

Examine the following code:

```
DECLARE
v_a BOOLEAN;
v_b BOOLEAN := FALSE;
v_c BOOLEAN ;
BEGIN
v_c := (v_a AND v_b);
-- Line A
....
END;
```

What is the value of V_C at Line A?
Mark for Review
(1) Points

True

False (*)

NULL

Undefined

Examine the following code:

```
DECLARE
v_score NUMBER(3);
v_grade CHAR(1);
BEGIN
v_grade := CASE v_score
-- Line A
....
```

The CASE expression must convert a numeric score to a letter grade: 90 -> A, 80 -> B, 70 -> C and so on. What should be coded at Line A?

Mark for Review
(1) Points

WHEN 90 THEN grade := 'A'

WHEN 90 THEN v_grade := 'A';

WHEN 90 THEN 'A' (*)

WHEN 90 THEN 'A';

Look at this code:

```
DECLARE
v_bool BOOLEAN := TRUE;
v_date DATE;
BEGIN
LOOP
EXIT WHEN v_bool;
SELECT SYSDATE INTO v_date FROM dual;
END LOOP;
END;
```

How many times will the SELECT statement execute?
Mark for Review
(1) Points

Once.

Twice.

Never (the SELECT will not execute at all) (*)

An infinite number of times because the EXIT condition will never be true

Incorrect Incorrect. Refer to Section 4.

Examine the following code:

```
DECLARE
v_count NUMBER := 0;
v_string VARCHAR2(20);
BEGIN
LOOP
v_string := v_string || 'x';
IF LENGTH(v_string) > 10 THEN
EXIT;
END IF;
v_count := v_count + 1;
END LOOP;
DBMS_OUTPUT.PUT_LINE(v_count);
END;
```

What will be displayed when this block is executed?
Mark for Review
(1) Points

9

10 (*)

11

xxxxxxxxxxx

You want to calculate and display the multiplication table for "sevens": 7x1=7, 7x2=14, 7x3=21 and so on. Which kind of PL/SQL construct is best for this? Mark for Review
(1) Points

A loop (*)

A CASE statement

IF ... END IF;

A Boolean variable.

PlSQL feedback midterm semister 1 part2

1.
Nonprocedural languages allow the programmer to produce a result when a series of steps are followed. True or False?
Mark for Review

True
False (*)

2. In which three ways does PL/SQL extend the SQL programming language? Mark for Review

(Choose all correct answers)

By adding procedural constructs. (*)

By adding compound constructs.

By adding iterative control. (*)

By adding conditional control. (*)

3. Which of the following statements is true? Mark for Review

You can embed PL/SQL statements within SQL code.

You can embed SQL statements within PL/SQL code. (*)

You can embed procedural constructs within SQL code.

None.

4. PL/SQL stands for: Mark for Review

Processing Language for SQL.

Procedural Language extension for SQL. (*)

Primary Language for SQL.

Proprietary Language for SQL.

PlSQL feedback midterm semister 1 part2

5. Which of the following statements is true? Mark for Review

PL/SQL is an Oracle proprietary, procedural, 3GL programming language. (*)

PL/SQL is an Oracle proprietary, procedural, 4GL programming language.

PL/SQL is an Oracle proprietary, nonprocedural, 3GL programming language.

PL/SQL is an ANSI-compliant, procedural programming language.

6. Which of the following statements about SQL is true? Mark for Review

SQL is an Oracle proprietary, nonprocedural, 4GL programming language.

SQL is an Oracle proprietary, procedural, 3GL programming language.

SQL is an ANSI-compliant, nonprocedural, 4GL programming language. (*)

SQL is an ANSI-compliant, procedural, 4GL programming language.

1. Which of the following can be compiled as a standalone program outside the database? Mark for Review
(1) Points

A program developed in PL/SQL

A program developed in Java

A program developed in C

All the above

Programs developed in Java or C, but not in PL/SQL. (*)

Incorrect Incorrect. Refer to Section 1.

1. You can create a Web site application written entirely in PL/SQL. True or False? Mark for Review
(1) Points

True (*)

False

Incorrect Incorrect. Refer to Section 1.
1. Procedural constructs give you better control of your SQL statements and their execution. True or False? Mark for Review
(1) Points

True (*)

False

Correct Correct
2. Which of the following can be compiled as a standalone program outside the database? Mark for Review
(1) Points

A program developed in PL/SQL

A program developed in Java

A program developed in C

All the above

Programs developed in Java or C, but not in PL/SQL. (*)

Incorrect Incorrect. Refer to Section 1.
3. PL/SQL differs from C and Java in which of the following ways? (Choose two.)
Mark for Review
(1) Points

(Choose all correct answers)

It requires an Oracle database or tool. (*)

It does not support object-oriented programming.

It is the most efficient language to use with an Oracle database. (*)

It is the most complex programming language to learn.

It is not portable to other operating systems.

Correct Correct

4. You can create a Web site application written entirely in PL/SQL. True or False? Mark for Review
(1) Points

True (*)

False

Incorrect Incorrect. Refer to Section 1.

5. When multiple SQL statements are combined into PL/SQL blocks, performance improves. True or False? Mark for Review
(1) Points

True (*)

False

Correct Correct

6. Which of the following can be done using PL/SQL? Mark for Review
(1) Points

Create complex applications.

Retrieve and modify data in Oracle database tables.

Manage database tasks such as security.

Create custom reports.

All of the above (*)

Incorrect Incorrect. Refer to Section 1.

1. Which of the following can be done using PL/SQL? Mark for Review
(1) Points

Create complex applications.

Retrieve and modify data in Oracle database tables.

Manage database tasks such as security.

Create custom reports.

All of the above (*)

Incorrect Incorrect. Refer to Section 1.

2. PL/SQL differs from C and Java in which of the following ways? (Choose two.) Mark for Review
(1) Points

(Choose all correct answers)

It requires an Oracle database or tool. (*)

It does not support object-oriented programming.

It is the most efficient language to use with an Oracle database. (*)

It is the most complex programming language to learn.

It is not portable to other operating systems.

Correct Correct

3. Procedural constructs give you better control of your SQL statements and their execution. True or False? Mark for Review
(1) Points

True (*)

False

Incorrect Incorrect. Refer to Section 1.

4. You can create a Web site application written entirely in PL/SQL. True or False? Mark for Review
(1) Points

True (*)

False

Correct Correct

5. Which of the following can be compiled as a standalone program outside the database? Mark for Review

PlSQL feedback midterm semister 1 part2

(1) Points

A program developed in PL/SQL

A program developed in Java

A program developed in C

All the above

Programs developed in Java or C, but not in PL/SQL. (*)

Incorrect Incorrect. Refer to Section 1.

6. When multiple SQL statements are combined into PL/SQL blocks, performance improves. True or False? Mark for Review

(1) Points

True (*)

False

Incorrect Incorrect. Refer to Section 1.

1. Which lines of code will correctly display the message "Hello World" ? (Choose two.) Mark for Review

(1) Points

(Choose all correct answers)

DBMS_OUTPUT('Hello World');

DBMS_OUTPUT.PUT_LINE('Hello World'); (*)

DBMS_OUTPUT.PUT_LINE('Hello' || 'World');

DBMS_OUTPUT.PUT_LINE('Hello' || ' ' || 'World'); (*)

Incorrect Incorrect. Refer to Section 1.

9. What are the characteristics of an anonymous block? (Choose two.) Mark for Review

(1) Points

(Choose all correct answers)

Unamed (*)

Stored in the database

Compiled each time the application is executed (*)

Can be declared as procedures or as functions

Correct Correct
10. Which of the following is NOT a PL/SQL programming environment? Mark for Review
(1) Points

Oracle jDeveloper

SQL*Plus

gSQL*Plus (*)

SQL Workshop in Application Express

Correct Correct
3. Which statements are mandatory in a PL/SQL block? (Choose two.) Mark for Review
(1) Points

(Choose all correct answers)

DECLARE

BEGIN (*)

EXCEPTION

END; (*)

Incorrect Incorrect. Refer to Section 1.
4. In a PL/SQL block, which of the following should not be followed by a semicolon? Mark for Review
(1) Points

DECLARE (*)

END

All SQL statements

All PL/SQL statements

Incorrect Incorrect. Refer to Section 1.

5. What is wrong with this PL/SQL anonymous block?

```
BEGIN
    DBMS_OUTPUT.PUT_LINE('Hello');
    DBMS_OUTPUT.PUT_LINE(' and Goodbye');
```

Mark for Review
(1) Points

The Declaration section is missing

The Exception section is missing

There is nothing wrong with the block, it will work fine.

The END; statement is missing (*)

Incorrect Incorrect. Refer to Section 1.

6. Which of the following is NOT a PL/SQL programming environment? Mark for Review
(1) Points

Oracle jDeveloper

SQL*Plus

gSQL*Plus (*)

SQL Workshop in Application Express

Correct Correct

7. How can you display results to check that a PL/SQL block is working correctly? Mark for Review
(1) Points

You don't need to do anything, the results will display automatically.

Use an Exception section

Use DBMS_OUTPUT.PUT_LINE (*)

Write a C or Java program to display the results

Incorrect Incorrect. Refer to Section 1.
10. Which of the following is a PL/SQL programming environment? Mark for Review
(1) Points

Oracle Cdeveloper

Java*Plus

PL/SQL Express

SQL*Workshop in Application Express (*)

Incorrect Incorrect. Refer to Section 1.
1. Which statement would select salaries that are greater than or equal to 2500 and less than or equal to 3500? Choose two correct answers. Mark for Review
(1) Points

(Choose all correct answers)

WHERE salary >= 2500 AND salary <= 3500 (*)

WHERE salary <=2500 AND salary >= 3500

WHERE salary BETWEEN 2500 AND 3500 (*)

WHERE BETWEEN salary = 2500 AND salary = 3500

Correct Correct
2. The F_FOOD_ITEMS table contains the FOOD_ITEM_NUMBER and the REGULAR_CODE columns. Which statement would display the FOOD_ITEM_NUMBER joined with the REGULAR_CODE without any space in between them? Mark for Review
(1) Points

SELECT food_item_number ' ' regular_code
FROM f_food_items;

SELECT food_item_number UNION regular_code
FROM f_food_items;

SELECT food_item_number || regular_code
FROM f_food_items;

(*)

SELECT food_item_numberregularcode
FROM f_food_items;

Incorrect Incorrect. Refer to Section 1.

3. The concatenation operator ... Mark for Review
(1) Points

Brings columns or character strings together

Creates a resultant column that is a character expression

Is represented by two vertical bars (||)

All of the above (*)

Incorrect Incorrect. Refer to Section 1.

4. Which of the following statements lists each employee's employee_id, salary, and salary plus a 20 percent bonus? Mark for Review
(1) Points

SELECT emp_id, salary, salary*.2
FROM employees;

SELECT emp_id, salary, salary*1.2
FROM employees;

(*)

SELECT emp_id, salary, salary*.8
FROM employees;

SELECT emp_id, salary, salary*20
FROM employees;

Incorrect Incorrect. Refer to Section 1.

5. Which of the following statements will generate a sentence such as the following:
The national holiday for United Arab Emirates is Independence Day.
for every country in the WF_COUNTRIES table?

Mark for Review
(1) Points

SELECT 'The national holiday for '|| country_name || ' is ' ||
national_holiday_name
FROM wf_countries;

SELECT "The national holiday for "|| country_name || " is " ||
national_holiday_name || "."
FROM wf_countries;

SELECT 'The national holiday for '|| country_name || ' is ' ||
national_holiday_name || '.'
FROM wf_countries;

(*)

SELECT 'The national holiday for || country_name || is ||
national_holiday_name || .'
FROM wf_countries;

Incorrect Incorrect. Refer to Section 1.

6. When using the LIKE operator, the "%" and "_" symbols can be used to do a pattern-matching, wild card search. True or False? Mark for Review
(1) Points

True (*)

False

Incorrect Incorrect. Refer to Section 1.

7. Examine the following statement:

```
SELECT country_name, population, population*.01
FROM wf_countries;
```

How would you modify this statement to display "Country", "Population", and "Expected Growth" as the column headings?
Mark for Review
(1) Points

SELECT country_name "COUNTRY", population "POPULATION", population*.01
"EXPECTED GROWTH"
FROM wf_countries;

(*)

SELECT country_name COUNTRY, population POPULATION, population*.01 EXPECTED

```
GROWTH
FROM wf_countries;
```

```
        SELECT country_name 'COUNTRY', population 'POPULATION', population*.01
'EXPECTED GROWTH'
FROM wf_countries;
```

```
        SELECT country_name, population, population*.01
FROM wf_countries
AS "COUNTRY", "POPULATION", "EXPECTED GROWTH";
```

Incorrect Incorrect. Refer to Section 1.

8. What SQL statement will return the ID, name, and area of all countries in the WF_COUNTRIES table, listed in order of greatest area to least area? Mark for Review
(1) Points

```
        SELECT country_id, country_name, area
FROM wf_countries
ORDER BY area DESC;
```

(*)

```
        SELECT country_id, country_name, area
FROM wf_countries
ORDER BY area ASC;
```

```
        SELECT country_id, country_name, area
FROM wf_countries
ORDER BY country_name;
```

```
        SELECT country_id, country_name, area
FROM wf_countries
GROUP BY area; pr />
```

Incorrect Incorrect. Refer to Section 1.

9. Which of the following statements diplays the population of the Republic of Benin (country_id 229) after a 3 percent growth in its population? Mark for Review
(1) Points

```
        SELECT country_name, population*.03
FROM wf_countries
WHERE country_id=229;
```

```
        SELECT country_name, population*1.03
FROM wf_countries
WHERE country_id=229;
```

(*)

```
        SELECT country_name, population*30
FROM wf_countries
WHERE country_id=229;
```

```
        SELECT country_name, population+population*.3
FROM wf_countries
WHERE country_id=229;
```

Incorrect Incorrect. Refer to Section 1.

10. Which of the following statements will display a sentence such as the following:
Aruba has an area of 193.
for every country in the WF_COUNTRIES table? Mark for Review
(1) Points

```
        SELECT country_name || ' has an area of ' || area
FROM wf_countries;
```

```
        SELECT country_name || 'has an area of' || area
FROM wf_countries;
```

```
        SELECT country_name || ' has an area of ' || area || '.'
FROM wf_countries;
```

(*)

```
        SELECT country_name " has an area of " area "."
FROM wf_countries;
```

Incorrect Incorrect. Refer to Section 1.

11. Which statement would display the departments in the EMPLOYEES table without displaying any duplicates? Mark for Review
(1) Points

```
        SELECT ALL department_id
FROM employees;
```

```
        SELECT department_id
FROM employees;
```

SELECT department_id
FROM employees
having ROWID=1;

SELECT DISTINCT department_id
FROM employees;

(*)

Incorrect Incorrect. Refer to Section 1.
12. If you want to SELECT all the columns of data in a table, you use which of the following symbols? Mark for Review
(1) Points

&

%

$

* (*)

Incorrect Incorrect. Refer to Section 1.
13. What can you use to change the column heading of calculated values in a SQL statement? Mark for Review
(1) Points

Multiplication operator

Column alias (*)

Concatenation operator

The DISTINCT keyword

Incorrect Incorrect. Refer to Section 1
1. What does the following SQL SELECT statement return?

SELECT UPPER(SUBSTR('Database Programming', INSTR('Database Programming','P'),20))
FROM dual;
Mark for Review
(1) Points

Programming

PROGRAMMING (*)

Database

DATABASE

Correct Correct

2. What function would you use to return the highest date in a month? Mark for Review
(1) Points

FINAL_DAY

END_DAY

HIGHEST_DAY

LAST_DAY (*)

Incorrect Incorrect. Refer to Section 1.

3. Which query would return a whole number if today's date is 26-MAY-04? Mark for Review
(1) Points

SELECT TRUNC(MONTHS_BETWEEN(SYSDATE,'19-MAR-79') /12)
AS YEARS
FROM DUAL;

(*)

SELECT TRUNC(YEARS_BETWEEN(SYSDATE,'19-MAR-79') /12)
AS YEARS
FROM DUAL;

SELECT MONTHS_BETWEEN(SYSDATE,'19-MAR-79') /12
AS YEARS
FROM DUAL;

None of the above

Incorrect Incorrect. Refer to Section 1.

4. Assume that today is December 31, 2007. What would be the output of the

following statement?

```
SELECT TO_CHAR(SYSDATE, 'DD/MM/Y') FROM DUAL;
```

Mark for Review
(1) Points

12/31/7

31-12-07

31/12/2007

31/12/7 (*)

Incorrect Incorrect. Refer to Section 1.

5. The following SQL statement will display the value: 456. True or False?

```
SELECT TRUNC(ROUND(456.98))
FROM dual;
```

Mark for Review
(1) Points

True

False (*)

Correct Correct

6. Which statement returns a user password combining the ID of an employee and the first 4 characters of their last name? Mark for Review
(1) Points

```
SELECT CONCAT (employee_id, SUBSTR(last_name,4,1))
AS "User Passwords"
FROM employees;
```

```
SELECT CONCAT (employee_id, INSTR(last_name,4,1))
AS "User Passwords"
FROM employees;
```

```
SELECT CONCAT (employee_id, INSTR(last_name,1,4))
AS "User Passwords"
FROM employees;
```

```
SELECT CONCAT (employee_id, SUBSTR(last_name,1,4))
AS "User Passwords"
FROM employees;
```

(*)

Incorrect Incorrect. Refer to Section 1.
7. Which of the following is not a number function? Mark for Review
(1) Points

TO_DATE (*)

ROUND

MOD

TRUNC

Incorrect Incorrect. Refer to Section 1.
8. Assume that today is January 10, 2008. What would be the output of the following statement?

```
SELECT TO_CHAR(SYSDATE, 'ddth "of" Month, YYYY') FROM DUAL;
```

Mark for Review
(1) Points

10th of January, 2008 (*)

10 January, 2008

10-January-2008

January 10th, 2008

Incorrect Incorrect. Refer to Section 1.
9. NULL means the same thing as a space or 0 (zero). True or False? Mark for Review
(1) Points

True

False (*)

Correct Correct
10. Which SQL statement will display each country's name with the first letter (only) of each word in uppercase? Mark for Review
(1) Points

```
        SELECT UPPER(country_name)
FROM wf_countries;



        SELECT lower(country_name)
FROM wf_countries;



        SELECT INITCAP(country_name)
FROM wf_countries;
```

(*)

```
        SELECT country_name
FROM wf_countries
ORDER BY INITCAP(country_name);
```

Incorrect Incorrect. Refer to Section 1.
11. What is returned by the following statement?
SELECT CONCAT('Today is','Thursday!') FROM DUAL; Mark for Review
(1) Points

TodayisThursday!

Today isThursday! (*)

today is thursday!

Today is Thursday!

Incorrect Incorrect. Refer to Section 1.
12. Which function compares two expressions? Mark for Review
(1) Points

NVL

NULLIF (*)

NVL2

NULL

Incorrect Incorrect. Refer to Section 1.
1. After they are declared, variables can be used only once in an application.

PlSQL feedback midterm semister 1 part2

True or False? Mark for Review
(1) Points

True

False (*)

Correct Correct

2. A function called FORMAT_TODAYS_DATE accepts no parameters and returns today's date in the format: Month DD, YYYY
The following anonymous block invokes the function:

```
DECLARE v_today DATE; BEGIN -- invoke the function here
```

Which of the following statements correctly assigns the date variable v_today to the value returned by the format_todays_date function?
Mark for Review
(1) Points

format_todays_date := v_today('Month DD, YYYY');

v_today := format_todays_date ('Month DD, YYYY');

v_today := format_todays_date(v_today);

v_today := TO_DATE(format_todays_date, 'Month DD, YYYY'); (*)

Incorrect Incorrect. Refer to Section 2.

3. Evaluate the following declaration. Determine whether or not it is legal.

```
DECLARE
    name,dept VARCHAR2(14);
```

Mark for Review
(1) Points

legal

illegal (*)

Correct Correct

4. Evaluate the following declaration. Determine whether or not it is legal.

```
DECLARE
    test NUMBER(5);
```

Mark for Review
(1) Points

legal (*)

illegal

Correct Correct

5. Which of the following are required when declaring a variable? (Choose two.) Mark for Review

(1) Points

(Choose all correct answers)

Identifier name (*)

CONSTANT

Data type (*)

NOT NULL

Correct Correct

6. Constants must be initialized. True or False? Mark for Review

(1) Points

True (*)

False

Incorrect Incorrect. Refer to Section 2.

7. Examine the following variable declarations:

DECLARE v_number NUMBER := 10; v_result NUMBER;

Which of the following correctly assigns the value 50 to V_RESULT? Mark for Review

(1) Points

v_result := v_number * 5;

v_result := 100 / 2;

v_result := ROUND(49.77);

All of the above. (*)

Incorrect Incorrect. Refer to Section 2.

1. Which of the following symbols can be used to enclose a comment in PL/SQL? Mark for Review

(1) Points

? ?

*/ / *

:: ::

/* */ (*)

Incorrect Incorrect. Refer to Section 2.
2. The name of a variable is an example of an identifier. True or False? Mark for Review
(1) Points

True (*)

False

Correct Correct
3. What is a lexical unit? Mark for Review
(1) Points

A data type for a column

A building block of a PL/SQL block (*)

A type of variable

Correct Correct
4. Which of the following are lexical units? (Choose two.) Mark for Review
(1) Points

(Choose all correct answers)

Data types

PL/SQL blocks

Identifiers (*)

Literals (*)

Incorrect Incorrect. Refer to Section 2.
5. Which of the following is a valid naming convention for an identifier? (Choose two.) Mark for Review
(1) Points

(Choose all correct answers)

Can include letters or numbers (*)

Cannot contain a reserved word (*)

Can be over 30 characters

Can start with a number or special character

Incorrect Incorrect. Refer to Section 2.
6. What characters must enclose non-numeric literal values? Mark for Review
(1) Points

Double quotes: " "

Parentheses: ()

Single quotes: ' ' (*)

Incorrect Incorrect. Refer to Section 2.
1. A datatype specifies and restricts the possible data values that can be assigned to a variable. True or False? Mark for Review
(1) Points

True (*)

False

Incorrect Incorrect. Refer to Section 2.
2. A Scalar data type holds a ____ value. Mark for Review
(1) Points

Multi

Large

Single (*)

Incorrect Incorrect. Refer to Section 2.
3. What are the data types of the variables in the following declaration?

```
DECLARE
fname VARCHAR2(20);
fname VARCHAR2(15) DEFAULT 'fernandez';
BEGIN
...
```

Mark for Review
(1) Points

Scalar (*)

Composite

LOB

Correct Correct
4. Which of the following is a composite data type? Mark for Review
(1) Points

CLOB

VARCHAR2

RECORD (*)

DATE

Correct Correct
5. Which of the folowing are scalar data types? (Choose three.) Mark for Review
(1) Points

(Choose all correct answers)

Array

Character (*)

Table

Date (*)

PlSQL feedback midterm semister 1 part2

Boolean (*)

Incorrect Incorrect. Refer to Section 2.

5. Which of the folowing are scalar data types? (Choose three.) Mark for Review
(1) Points

(Choose all correct answers)

Array

Character (*)

Table

Date (*)

Boolean (*)

Incorrect Incorrect. Refer to Section 2.

6. Which of the following are PL/SQL data types? (Choose three.) Mark for Review
(1) Points

(Choose all correct answers)

Large Objects (LOB) (*)

Lexical

Scalar (*)

Delimiter

Composite (*)

Incorrect Incorrect. Refer to Section 2.

1. If you use the %TYPE attribute, you can avoid hard-coding the column name. True or False? Mark for Review
(1) Points

True

False (*)

Correct Correct
2. Which of the following is NOT a character data type? Mark for Review
(1) Points

VARCHAR2

BOOLEAN (*)

CHAR

LONG

Correct Correct
3. When declared using %TYPE, a variable will inherit ____ from the column on which it is based. Mark for Review
(1) Points

The name of the column

The value of the column

The data type and size of the column (*)

Correct Correct
4. Which of the following is NOT a good guideline for declaring variables? Mark for Review
(1) Points

Declare one identifier per line

Use column names as identifiers (*)

Use NOT NULL when the variable must have a value

Correct Correct
5. Code is easier to read if you declare one identifier per line. True or False? Mark for Review
(1) Points

True (*)

False

Correct Correct
6. Which of the following variable declarations does NOT use a number data type? Mark for Review
(1) Points

v_count PLS_INTEGER := 0;

v_median_age NUMBER(6,2);

v_students LONG; (*)

v_count BINARY_INTEGER;

Correct Correct
1. When a join condition is omitted completely the result is a Cartesian product in which all combinations of rows will be displayed. True or False? Mark for Review
(1) Points

True (*)

False

Incorrect Incorrect. Refer to Section 2.
2. A nonequijoin combines tables that have one or more exact matching columns. True or False? Mark for Review
(1) Points

True

False (*)

Incorrect Incorrect. Refer to Section 2.
3. What kind of join is used in the following example?

```
SELECT e.employee_id, e.last_name, j.grade_level
FROM employees e, job_grades j
WHERE e.salary BETWEEN j.lowest_sal and j.highest_sal;
```

Mark for Review
(1) Points

Simple join

Equijoin

Nonequijoin (*)

Outer join

Correct Correct
4. Table aliases can be used to shorten the syntax in join statements. True or False? Mark for Review
(1) Points

True (*)

False

Incorrect Incorrect. Refer to Section 2.
5. Will the following statement execute correctly?

```
SELECT department_id, department_name, last_name
FROM employees e, departments d
WHERE e.department_id = d.department_id;
```

Mark for Review
(1) Points

Yes, there are no errors in this statement.

No, because one column has been ambiguously defined. (*)

No, because every column must be prefixed by its table alias, for example: e.last_name.

Yes, Oracle will resolve which department_id column comes from which table.

Correct Correct
6. What type of join returns rows for one table even when there are no matching rows in the other table? Mark for Review
(1) Points

Simple join

Equijoin

Nonequijoin

Outer join (*)

Incorrect Incorrect. Refer to Section 2.
7. What does the following statement return?

```
SELECT e.last_name, d.department_id, d.department_name
FROM employees e, departments d
WHERE e.department_id(+) = d.department_id
ORDER BY e.department_id;
```

Mark for Review
(1) Points

Returns all departments, even if there are no employees in the department. (*)

Returns all employees, even if they have not been assigned to a department.

Returns only those departments that contain at least one employee

Returns all possible combinations of employees and departments.

Correct Correct
8. If table A has 20 rows and table B has 10 rows, how many rows will be returned if you perform a Cartesian product on those two tables? Mark for Review
(1) Points

20

10

200 (*)

120

Correct Correct
1. The following EMPLOYEE_ID, SALARY, and COMMISSION_PCT data in the EMPLOYEES table for six employees.

DATA: 143, 2600, null
144, 2500, null
149, 10500, .2
174, 11000, .3
176, 8600, .2
178, 7000, .15

What is the result of the following statement:

```
SELECT AVG(commission_pct)
FROM employees
WHERE employee_id IN( 143,144,149,174,176,178)
```

Mark for Review

(1) Points

0.1416

0.2125 (*)

The statement will fail because you cannot use more than one group function in a single statement.

0.2521

Correct Correct
2. What will be returned when the following statement is executed?
SELECT last_name
FROM employees
WHERE salary > ALL
(SELECT salary FROM employees
WHERE job_id = 'IT_PROG');
Mark for Review
(1) Points

The names of all IT Programmers.

The names of employees who earn more than every IT Programmer. (*)

The names of employees who earn more than at least one IT Programmer.

The names of employees who earn more than half of the IT Programmers.

Incorrect Incorrect. Refer to Section 2.
3. When using a subquery, the =ANY and IN operators are logically identical; they will always give the same result as each other. True or False? Mark for Review
(1) Points

True (*)

False

Incorrect Incorrect. Refer to Section 2.
4. Read the following SELECT statement. Choose the column or columns that MUST be included in the GROUP BY clause.

SELECT region_id, COUNT(country_id)
FROM wf_countries
GROUP BY ?????
Mark for Review

(1) Points

region_id, COUNT(country_id)

region_id,country_id

country_id

region_id (*)

Incorrect Incorrect. Refer to Section 2.

5. Single row subqueries may NOT include which of these operators? Mark for Review
(1) Points

ALL (*)

=

<>

>

Incorrect Incorrect. Refer to Section 2.

6. Which of the following SQL statements will display the name and a total of people with the same last name? Mark for Review
(1) Points

SELECT last_name, COUNT(employee_id)
FROM EMPLOYEES
GROUP BY last_name;

(*)

SELECT employee_id, COUNT(last_name)
FROM EMPLOYEES
GROUP BY last_name;

SELECT last_name, DISTINCT COUNT(employee_id)
FROM EMPLOYEES
GROUP BY last_name;

SELECT employee_id, DISTINCT(last_name)
FROM EMPLOYEES
GROUP BY last_name;

Incorrect Incorrect. Refer to Section 2.
7. Group functions cannot be used in subqueries because they contain too many rows. True or False? Mark for Review
(1) Points

True

False (*)

Correct Correct
8. What would the following SQL statement return?
SELECT MAX(hire_date) FROM employees; Mark for Review
(1) Points

The hire date of the longest serving employee.

The hire date of the newest (most recently hired) employee. (*)

The hire dates of all employees in ascending order.

The hire dates of all employees.

Correct Correct
1. Which of the following is correct? Mark for Review
(1) Points

v_family_name = SMITH;

V_FAMILY_NAME = SMITH;

v_family_name := SMITH;

v_family_name := 'SMITH'; (*)

Incorrect Incorrect. Refer to Section 2.
2. When PL/SQL converts data automatically from one data type to another, it is called _____ conversion. Mark for Review
(1) Points

Explicit

Implicit (*)

TO_CHAR

Correct Correct
3. The DECODE and MAX functions can be used in PL/SQL statements. True or False? Mark for Review
(1) Points

True

False (*)

Correct Correct
4. Examine the following code: DECLARE x VARCHAR2(20); BEGIN x:= 5 + 4 * 5 ; DBMS_OUTPUT.PUT_LINE(x); END; What value of x will be displayed? Mark for Review
(1) Points

45

29

25 (*)

14

Incorrect Incorrect. Refer to Section 2.
5. Which of the following statements about implicit conversions is NOT true? Mark for Review
(1) Points

Code containing implicit conversions typically runs faster than code containing explicit conversions. (*)

Code containing implicit conversions may not work in the future if Oracle changes the conversion rules.

Code containing implicit conversions is harder to read and understand.

Incorrect Incorrect. Refer to Section 2.
6. The LENGTH and ROUND functions can be used in PL/SQL statements. True or False? Mark for Review
(1) Points

True (*)

False

Incorrect Incorrect. Refer to Section 2.
7. Which of the following data type conversions can be done implicitly? (Choose two.) Mark for Review
(1) Points

(Choose all correct answers)

DATE to NUMBER

NUMBER to VARCHAR2 (*)

NUMBER to PLS_INTEGER (*)

Incorrect Incorrect. Refer to Section 2.
8. Which of the following are valid PL/SQL operators? (Choose three.) Mark for Review
(1) Points

(Choose all correct answers)

Concatenation (*)

Exception

Exponential (*)

Arithmetic (*)

Incorrect Incorrect. Refer to Section 2.
9. What will happen when the following code is executed?

```
DECLARE v_new_date DATE;
BEGIN
v_new_date := 'Today';
DBMS_OUTPUT.PUT_LINE(v_new_date);
END;
```

Mark for Review
(1) Points

The block will execute and display today's date.

The block will execute and display the word "Today".

The block will fail because the character value "Today" cannot be implicitly converted to a date. (*)

Incorrect Incorrect. Refer to Section 2
PL/SQL statements must be written on a single line. Mark for Review
(1) Points

True

False (*)

Correct Correct
Which explicit function is used to convert a character into a number? Mark for Review
(1) Points

TO_DATE

TO_NUMBER (*)

TO_CHAR

Incorrect Incorrect. Refer to Section 2.
Examine the following block. What should be coded at Line A?

```
DECLARE
v_char VARCHAR2(8) := '24/09/07';
v_date DATE;
BEGIN
v_date := ..... Line A
END;
```

Mark for Review
(1) Points

v_date := FROM_CHAR(v_char,'dd/mm/yy');

v_date := TO_DATE(v_char,'dd/mm/yy'); (*)

v_date := v_char;

Correct Correct
PL/SQL can implicitly convert a CHAR to a NUMBER, provided the CHAR contains a numeric value, for example '123'. True or False? Mark for Review
(1) Points

True (*)

False

Incorrect Incorrect. Refer to Section 2.
Using implicit conversions is good programming practice. Mark for Review
(1) Points

True

False (*)

Correct Correct
The TO_CHAR function is used for explicit data type conversions. True or False?
Mark for Review
(1) Points

True (*)

False

Correct Correct
1. Examine the following code: DECLARE x VARCHAR2(20); BEGIN x:= 5 + 4 * 5 ; DBMS_OUTPUT.PUT_LINE(x); END; What value of x will be displayed? Mark for Review
(1) Points

45

29

25 (*)

14

Correct Correct
2. What will happen when the following code is executed?

```
DECLARE v_new_date DATE;
BEGIN
v_new_date := 'Today';
DBMS_OUTPUT.PUT_LINE(v_new_date);
END;
```

Mark for Review
(1) Points

The block will execute and display today's date.

The block will execute and display the word "Today".

The block will fail because the character value "Today" cannot be implicitly converted to a date. (*)

Incorrect Incorrect. Refer to Section 2.

3. Which of the following data type conversions can be done implicitly? (Choose two.) Mark for Review
(1) Points

(Choose all correct answers)

DATE to NUMBER

NUMBER to VARCHAR2 (*)

NUMBER to PLS_INTEGER (*)

Incorrect Incorrect. Refer to Section 2.

4. Using implicit conversions is good programming practice. Mark for Review
(1) Points

True

False (*)

Correct Correct

5. Which of the following are valid PL/SQL operators? (Choose three.) Mark for Review
(1) Points

(Choose all correct answers)

Concatenation (*)

Exception

Exponential (*)

Arithmetic (*)

Incorrect Incorrect. Refer to Section 2.

6. PL/SQL can implicitly convert a CHAR to a NUMBER, provided the CHAR contains a numeric value, for example '123'. True or False? Mark for Review

(1) Points

True (*)

False

Incorrect Incorrect. Refer to Section 2.

7. Which explicit function is used to convert a character into a number? Mark for Review

(1) Points

TO_DATE

TO_NUMBER (*)

TO_CHAR

Correct Correct

8. Examine the following block. What should be coded at Line A?

```
DECLARE
v_char VARCHAR2(8) := '24/09/07';
v_date DATE;
BEGIN
v_date := ..... Line A
END;
```

Mark for Review

(1) Points

v_date := FROM_CHAR(v_char,'dd/mm/yy');

v_date := TO_DATE(v_char,'dd/mm/yy'); (*)

v_date := v_char;

Incorrect Incorrect. Refer to Section 2.

9. When PL/SQL converts data automatically from one data type to another, it is called _____ conversion. Mark for Review

(1) Points

Explicit

Implicit (*)

TO_CHAR

Correct Correct
10. The LENGTH and ROUND functions can be used in PL/SQL statements. True or False? Mark for Review
(1) Points

True (*)

False

Incorrect Incorrect. Refer to Section 2.
11. The TO_CHAR function is used for explicit data type conversions. True or False? Mark for Review
(1) Points

True (*)

False

Correct Correct
12. PL/SQL statements must be written on a single line. Mark for Review
(1) Points

True

False (*)

Correct Correct
13. Which of the following is correct? Mark for Review
(1) Points

v_family_name = SMITH;

V_FAMILY_NAME = SMITH;

v_family_name := SMITH;

v_family_name := 'SMITH'; (*)

Incorrect Incorrect. Refer to Section 2.
14. The DECODE and MAX functions can be used in PL/SQL statements. True or False? Mark for Review

(1) Points

True

False (*)

Correct Correct
15. Which of the following statements about implicit conversions is NOT true?
Mark for Review
(1) Points

Code containing implicit conversions typically runs faster than code containing explicit conversions. (*)

Code containing implicit conversions may not work in the future if Oracle changes the conversion rules.

Code containing implicit conversions is harder to read and understand.

Incorrect Incorrect. Refer to Section 2.
1. What values will be displayed when the following code is executed?

```
DECLARE
    v_mynum NUMBER;
BEGIN
    v_mynum := 7;
    DECLARE
       v_mynum NUMBER;
    BEGIN
       DBMS_OUTPUT.PUT_LINE(v_mynum);
       v_mynum := 3;
    END;
    DBMS_OUTPUT.PUT_LINE(v_mynum);
END;
```

Mark for Review
(1) Points

3,3

3,7

Null, 7 (*)

Null, 3

Incorrect Incorrect. Refer to Section 2.
2. What happens when an exception occurs in the executable section of a PL/SQL block? Mark for Review

(1) Points

Oracle keeps trying to re-execute the statement which caused the exception.

The remaining statements in the executable section are not executed. Instead, Oracle looks for an EXCEPTION section in the block. (*)

The remaining statements in the executable section of the block are executed.

The exception is always propagated to the calling environment.

Incorrect Incorrect. Refer to Section 2.

3. Examine the following code. At Line A, we want to assign a value of 25 to the outer block's variable (V1). What must we do?

```
DECLARE
    v_myvar NUMBER; -- This is V1
BEGIN
    DECLARE
       v_myvar NUMBER := 8;
       BEGIN
          -- Line A
       END;
END;
```

Mark for Review
(1) Points

At Line A, code:
v_myvar := 25;

Label both blocks and at line A, code:
v_myvar := 25;

It cannot be done because the outer block's v_myvar is out of scope at Line A.

Label the outer block and (at Line A) dot-prefix v_myvar with the block label.

(*)

It cannot be done because the outer block's v_myvar is in scope but not visible at Line A.

Incorrect Incorrect. Refer to Section 2.

4. An inner block is nested within an outer block. An exception occurs within the inner block, but the inner block does not have an EXCEPTION section. What happens? Mark for Review
(1) Points

The exception is propagated to the outer block and the remaining executable statements in the outer block are skipped. (*)

The exception is propagated to the outer block and the remaining executable statements in the outer block are executed.

Oracle automatically tries to re-execute the inner block.

The outer block is bypassed and the exception is always propagated to the calling environment.

Correct Correct

5. What is wrong with this code?

```
DECLARE
    v_a NUMBER;
BEGIN
    v_a := 27;
    <<inner_block>>
    BEGIN
       v_a := 15;
END;
```

Mark for Review
(1) Points

The outer block has no label.

Variable v_a is out of scope within the inner block and therefore cannot be referenced.

The inner block has no END; statement. (*)

Nothing is wrong, the code will execute successfully.

Correct Correct

6. Examine the following code. What is the scope of variable v_myvar?

```
DECLARE
    v_myvar NUMBER;
BEGIN
    v_myvar := 6;
    DECLARE
       v_hervar NUMBER;
    BEGIN
       v_hervar := 4;
    END;
END;
```

Mark for Review

(1) Points

Only the outer block

Both the inner and the outer block (*)

Only the inner block

Neither block

Incorrect Incorrect. Refer to Section 2.

7. Examine the following nested blocks. Line B causes an exception. What will be displayed when this code is executed?

```
DECLARE
    var_1 NUMBER;
BEGIN
    var_1 := 4;
    DECLARE
       var_2 NUMBER;
    BEGIN
       var_2 := 'Unhappy'; -- Line B
       var_1 := 8;
    END;
    var_1 := 12;
EXCEPTION
    WHEN OTHERS THEN
       DBMS_OUTPUT.PUT_LINE(var_1);
END;
```

Mark for Review
(1) Points

Unhappy

12

8

4 (*)

Incorrect Incorrect. Refer to Section 2.

8. Examine the following code. Line A causes an exception. What will be displayed when the block is executed?

```
DECLARE
    x NUMBER := 10;
    y NUMBER;
BEGIN
    x := 15;
    y := 'Happy'; -- Line A
    x := 20;
```

```
EXCEPTION
    WHEN OTHERS THEN
    DBMS_OUTPUT.PUT_LINE(x);
END;
```

Mark for Review
(1) Points

10

20

15 (*)

Nothing is displayed

Incorrect Incorrect. Refer to Section 2.

1. What is wrong with the following statement?

```
DELETE from employees WHERE salary > (SELECT MAX(salary) FROM employees);
```

Mark for Review
(1) Points

You cannot code a subquery inside a DELETE statement.

You cannot use inequality operators such as "<" and ">" inside a DELETE statement.

Nothing is wrong, the statement will execute correctly. (*)

Incorrect Incorrect. Refer to Section 3.

2. To modify an existing row in a table, you can use the _____ statement.

Mark for Review
(1) Points

MODIFY

INSERT

ALTER

UPDATE (*)

Incorrect Incorrect. Refer to Section 3.

3. What is wrong with the following statement? MERGE INTO emps e USING new_emps ne ON (e.employee_id = ne.employee_id) WHEN MATCHED THEN UPDATE SET ne.salary = e.salary WHEN NOT MATCHED THEN INSERT VALUES (ne.employee_id, ne.first_name, ne.last_name, ... ne.salary, ...); Mark for Review

(1) Points

The UPDATE clause must include the target table name: UPDATE emps SET ....

The INSERT clause must include a column list as well as a list of column values.

The SET clause is trying to update the source table from the target table. (*)

Nothing is wrong, the statement will execute correctly.

Correct Correct

4. You want to modify existing rows in a table. Which of the following are NOT needed in your SQL statement? (Choose Two) Mark for Review
(1) Points

(Choose all correct answers)

A MODIFY clause (*)

An UPDATE clause

The name of the table

The name of the column(s) you want to modify.

A new value for the column you want to modify (this can be an expression or a subquery).

A WHERE clause. (*)

5. Is it possible to insert more than one row at a time using an INSERT statement with a VALUES clause? Mark for Review
(1) Points

No, you can only create one row at a time when using the VALUES clause. (*)

Yes, you can list as many rows as you want, just remember to separate the rows with commas.

No, there is no such thing as INSERT ... VALUES.

Incorrect Incorrect. Refer to Section 3.

What would be the result of the following statement: DELETE employees; Mark for Review
(1) Points

Nothing, no data will be changed.

All rows in the employees table will be deleted. (*)

The statement will fail because it contains a syntax error.

The row with EMPOYEE_ID=100 will be deleted.

Incorrect Incorrect. Refer to Section 3.
When inserting a row into a table, the VALUES clause must include a value for every column of the table. True or False? Mark for Review
(1) Points

True

False (*)

Correct Correct
Look at this SQL statement: MERGE INTO old_trans ot USING new_trans nt ON (ot.trans_id = nt.trans_id) .... ; OLD_TRANS is the source table and NEW_TRANS is the target table. True or false? Mark for Review
(1) Points

True

False (*)

Incorrect Incorrect. Refer to Section 3.
1. It is good programming practice to create identifiers having the same name as column names. True or False? Mark for Review
(1) Points

True

False (*)

Correct Correct
2. Look at this PL/SQL block: DECLARE v_count NUMBER; BEGIN SELECT COUNT(*) INTO v_count FROM employees WHERE salary > 50000; END; No employees earn more than $50000. Which of the following statements are true? (Choose two). Mark for Review
(1) Points

(Choose all correct answers)

The SELECT will return value 0 into V_COUNT. (*)

The SELECT will fail because it does NOT return exactly one row.

The block will fail because variable V_SALARY was not declared.

The SELECT returns exactly one row. (*)

The block will fail because no results are displayed to the user.

Incorrect Incorrect. Refer to Section 3.
Which of the following is NOT a valid guideline for retrieving data in PL/SQL? Mark for Review
(1) Points

Terminate the SQL statement with a semicolon (;)

Do NOT use a WHERE clause in SELECT statements. (*)

Where possible, declare variables using the %TYPE attribute.

Specify the same number of variables in the INTO clause as database columns in the SELECT clause.

Incorrect Incorrect. Refer to Section 3.
When used in a PL/SQL block, which SQL statement must return exactly one row? Mark for Review
(1) Points

INSERT

UPDATE

SELECT (*)

MERGE

DELETE

Correct Correct
5. Which SQL statements can be used directly in a PL/SQL block? (Choose two.)
Mark for Review

PlSQL feedback midterm semister 1 part2

(1) Points

(Choose all correct answers)

GRANT EXECUTE ON ...

SELECT * INTO ... (*)

REVOKE SELECT ON ...

UPDATE employees SET... (*)

ALTER TABLE employees ...

Incorrect Incorrect. Refer to Section 3.

6. Does PL/SQL allow you to have a variable with the same name as a database column? Mark for Review

(1) Points

No

Yes (*)

Correct Correct

7. What will happen when the following block is executed? DECLARE v_last employees.last_name%TYPE; v_first employees.first_name%TYPE; v_salary employees.salary%TYPE; BEGIN SELECT first_name, last_name INTO v_first, v_last, v_salary FROM employees WHERE employee_id=100; END; Mark for Review

(1) Points

The block will fail because the SELECT statement returns more than one row.

The block will fail because the SELECT is trying to read two columns into three PL/SQL variables. (*)

The block will fail because V_LAST was declared before V_FIRST.

The block will execute successfully, and the V_SALARY variable will be set to NULL.

Incorrect Incorrect. Refer to Section 3.

8. Which one of these SQL statements can be directly included in a PL/SQL executable block? Mark for Review

(1) Points

IF... THEN...;

INSERT INTO...; (*)

SELECT * FROM DUAL;

SHOW USER;

Incorrect Incorrect. Refer to Section 3.
1. Employee_id 999 does not exist. What will happen when the following code is executed? DECLARE employee_id employees.employee_id%TYPE := 999; BEGIN UPDATE employees SET salary = salary * 1.1 WHERE employee_id = employee_id; END; Mark for Review
(1) Points

No rows are updated but the block completes successfully.

Every employee row is updated. (*)

An exception is raised because you cannot give a variable the same name as a table column.

An exception is raised because the UPDATE statement did not modify any rows.

Correct Correct
2. A PL/SQL block contains the following DML statement: UPDATE wf_countries SET population = population * 1.1 WHERE country_id = 229; Which kind of cursor is used for this statement? Mark for Review
(1) Points

An implicit cursor named "WF_COUNTRIES".

An implicit cursor named "SQL". (*)

An explicit cursor named "SQL".

An explicit cursor which must be declared and named by the PL/SQL programmer.

Incorrect Incorrect. Refer to Section 3.
3. There are three employees in department 90. What will be displayed when the following code is executed? DECLARE v_open CHAR(3) := 'NO'; BEGIN UPDATE employees SET job_id = 'ST_CLERK' WHERE department_id = 90; IF SQL%FOUND THEN v_open := 'YES'; END IF; DBMS_OUTPUT.PUT_LINE(v_open || ' ' || SQL%ROWCOUNT); END; Mark for Review
(1) Points

NO 3

YES 1

YES 3 (*)

Nothing will be displayed. The block will fail because you cannot use implicit cursor attributes directly in a call to DBMS_OUTPUT.PUT_LINE.

Correct Correct
4. You can use implicit cursor attributes such as SQL%ROWCOUNT directly inside a DML statement. For example: INSERT INTO log_table VALUES (SYSDATE, USER, SQL%ROWCOUNT); True or False? Mark for Review
(1) Points

True

False (*)

Correct Correct
5. Which of the following use an implicit cursor? Mark for Review
(1) Points

DML statements only.

SELECT statements only.

DML statements and SELECT statements which return a single row. (*)

COMMIT and ROLLBACK statements only.

Correct Correct
6. Which of the following SQL DML commands can be used inside a PL/SQL block? Mark for Review
(1) Points

INSERT and UPDATE only.

UPDATE and DELETE only.

INSERT, UPDATE and DELETE only.

PlSQL feedback midterm semister 1 part2

INSERT, UPDATE, DELETE and MERGE. (*)

Correct Correct

1. How many INSERTs can you have in one transaction? Mark for Review
(1) Points

One

As many as you want until you do a COMMIT or ROLLBACK. (*)

As many as you can execute before the database does an AUTOSAVE.

As many as you want until a different DML statement (UPDATE, DELETE or MERGE) is executed.

Incorrect Incorrect. Refer to Section 3.

2. How many transactions are in the following block?

```
BEGIN
    INSERT INTO countries (country_id, country_name)
       VALUES ('XA', 'Xanadu');
    INSERT INTO countries (country_id, country_name)
       VALUES ('NV', 'Neverland');
    UPDATE countries SET country_name='Deutchland'
       WHERE country_id='DE';
    UPDATE countries SET region_id=1
       WHERE country_name LIKE '%stan';
END;
```

How many transactions are shown above?
Mark for Review
(1) Points

Two; both the INSERTs are one transaction and both the UPDATEs are a second transaction.

It depends on how many rows are updated - there will be a separate transaction for each row.

One (*)

Incorrect Incorrect. Refer to Section 3.

3. Examine the following code: BEGIN
INSERT INTO animals VALUES ('aa','aardvarks');
SAVEPOINT sp_1;
INSERT INTO animals VALUES ('bb','big birds');
SAVEPOINT sp_2;

```
ROLLBACK TO sp_1;
INSERT INTO animals VALUES ('cc','cool cats');
COMMIT;
END;
```

Which row(s) will be in the ANIMALS table after this block is executed? Mark for Review
(1) Points

cool cats

big birds and cool cats

aardvaarks and cool cats (*)

aardvaarks, big birds and cool cats

Correct Correct

4. In a PL/SQL block, where can you code a COMMIT statement? Mark for Review
(1) Points

In any section of the block: Declaration, Executable, or Exception.

Only the Executable section.

In the Executable and/or the Exception sections. (*)

Nowhere; the COMMIT statement must be outside the block.

Incorrect Incorrect. Refer to Section 3.

1. Which of the following statements are true about PL/SQL conditional control structures such as IF ... , CASE ... and loops? Mark for Review
(1) Points

They allow the programmer to use logical tests to determine which statements are executed and which are not.

They allow a set of statements to be executed repeatedly (i.e. more than once).

They determine a course of action based on conditions.

All of the above. (*)

Incorrect Incorrect. Refer to Section 4.

2. We want to execute one of three statements depending on whether the value in V_VAR is 10, 20 or some other value. What should be coded at Line A? IF v_var = 10 THEN statement1; -- Line A statement2; ELSE statement3; END IF; Mark for Review
(1) Points

ELSE IF v_var = 20 THEN

ELSIF v_var = 20

ELSIF v_var = 20 THEN (*)

IF v_var = 20 THEN

Incorrect Incorrect. Refer to Section 4.

3. What is wrong with the following trivial IF statement:

```
IF (v_job='President')
THEN v_salary := 10000;
```

Mark for Review
(1) Points

IF and THEN must be on the same line: IF (v_job='President') THEN ...

The condition should be coded: IF (v_job := 'President')

END IF; is missing (*)

ELSE is missing

Correct Correct

4. Which one of the following is correct syntax for an IF statement? Mark for Review
(1) Points

IF condition THEN DO statement1; statement2; END IF;

IF condition THEN statement1; statement2; END IF; (*)

IF condition THEN statement1; statement2; ENDIF;

IF condition THEN statement1; AND statement2; END IF;

Incorrect Incorrect. Refer to Section 4.

5. What will be displayed when this block is executed? DECLARE v_bool1 BOOLEAN

:= NULL; v_bool2 BOOLEAN := NULL; v_char VARCHAR(10) := 'Start'; BEGIN IF (v_bool1 = v_bool2) THEN v_char:='Equal'; ELSE v_char:='Not equal'; END IF; DBMS_OUTPUT.PUT_LINE(v_char); END;
Mark for Review
(1) Points

Equal

Not equal (*)

Start

Nothing will be displayed. The block will fail because you cannot compare two null values.

Incorrect
Incorrect. Refer to Section 4.

6. What will be displayed when this block is executed? DECLARE v_bool1 BOOLEAN := TRUE; v_bool2 BOOLEAN; v_char VARCHAR(4) := 'up'; BEGIN IF (v_bool1 AND v_bool2) THEN v_char:='down'; ELSE v_char:='left'; END IF; DBMS_OUTPUT.PUT_LINE(v_char); END;
Mark for Review
(1) Points

up

down

left (*)

null

Incorrect
Incorrect. Refer to Section 4.

7. Look at the following (badly written) code:

```
age := 5; IF age<30 THEN mature := 'adult';
ELSIF age<22 THEN mature := 'teenager';
ELSIF age<13 THEN mature := 'child';
END IF;
DBMS_OUTPUT.PUT_LINE(mature);
```

What will be displayed when this code is executed?
Mark for Review
(1) Points

child

teenager

adult (*)

adultteenagerchild

Incorrect Incorrect. Refer to Section 4.
8. You want to repeat a set of statements 100 times, incrementing a counter each time. What kind of PL/SQL control structure would you use? Mark for Review
(1) Points

IF...THEN...ELSE

IF...THEN...ELSIF...ELSE

CASE...WHEN...THEN

A loop. (*)

Correct Correct
1. Examine the following code:

```
DECLARE
v_a BOOLEAN;
v_b BOOLEAN := FALSE;
v_c BOOLEAN ;
BEGIN
v_c := (v_a AND v_b);
-- Line A
....
END;
```

What is the value of V_C at Line A?
Mark for Review
(1) Points

True

False (*)

NULL

Undefined

Incorrect Incorrect. Refer to Section 4.
2. Look at the following code:

```
DECLARE
x BOOLEAN := FALSE;
y BOOLEAN := FALSE;
z BOOLEAN ;
```

```
BEGIN
z := (x OR NOT y);
-- Line A
....
END;
```

What is the value of Z at Line A?
Mark for Review
(1) Points

True (*)

False

NULL

An error will occur because you cannot combine two Boolean variables using "NOT".

Incorrect Incorrect. Refer to Section 4.

3. What will be displayed when the following block is executed?

```
DECLARE
v_age1 NUMBER(3);
v_age2 NUMBER(3);
v_message VARCHAR2(20);
BEGIN
CASE
WHEN v_age1 = v_age2 THEN v_message := 'Equal';
WHEN v_age1 <> v_age2 THEN v_message := 'Unequal';
ELSE v_message := 'Undefined';
END CASE;
DBMS_OUTPUT.PUT_LINE(v_message);
END;
```

Mark for Review
(1) Points

Equal

Undefined (*)

Unequal

Nothing will be displayed because V_MESSAGE is set to NULL.

Incorrect Incorrect. Refer to Section 4.

4. Examine the following code:

```
DECLARE
v_score NUMBER(3);
```

```
v_grade CHAR(1);
BEGIN
v_grade := CASE v_score
-- Line A
....
```

The CASE expression must convert a numeric score to a letter grade: 90 -> A, 80 -> B, 70 -> C and so on. What should be coded at Line A?

Mark for Review
(1) Points

WHEN 90 THEN grade := 'A'

WHEN 90 THEN v_grade := 'A';

WHEN 90 THEN 'A' (*)

WHEN 90 THEN 'A';

Incorrect Incorrect. Refer to Section 4.

5. Examine the following code:

```
DECLARE
v_score NUMBER(3);
v_grade CHAR(1);
BEGIN
CASE v_score
-- Line A
....
```

The CASE statement must convert a numeric score to a letter grade: 90 -> A, 80 -> B, 70 -> C and so on.

What should be coded at Line A?

Mark for Review
(1) Points

WHEN 90 THEN v_grade := 'A'

WHEN 90 THEN v_grade := 'A'; (*)

WHEN 90 THEN 'A'

WHEN 90 THEN 'A';

Incorrect Incorrect. Refer to Section 4.

6. How must you end a CASE statement? Mark for Review
(1) Points

END;

END CASE; (*)

END IF;

ENDCASE;

Incorrect Incorrect. Refer to Section 4.

7. What will be displayed when the following block is executed?

```
DECLARE
v_age NUMBER(3);
v_gender VARCHAR2(6) := 'Female';
v_status VARCHAR2(20);
BEGIN
CASE
WHEN v_age >= 18 AND v_gender = 'Male' THEN v_status := 'Adult Male';
WHEN v_age >= 18 AND v_gender = 'Female' THEN v_status := 'Adult Female';
WHEN v_age < 18 AND v_gender = 'Male' THEN v_status := 'Junior Male';
WHEN v_age < 18 AND v_gender = 'Female' THEN v_status := 'Junior Female';
ELSE v_status := 'Other Value';
END CASE;
DBMS_OUTPUT.PUT_LINE(v_status);
END;
```

Mark for Review
(1) Points

Adult Male

Junior Female

Other Value (*)

Nothing will be displayed because V_STATUS is set to NULL.

Incorrect Incorrect. Refer to Section 4.

8. How must you end a CASE expression? Mark for Review
(1) Points

END; (*)

ENDIF;

END CASE;

ENDCASE;

Incorrect Incorrect. Refer to Section 4.
1. Which kind of loop is this?

```
i := 10;
LOOP
    i := i + 1;
    EXIT WHEN i > 30;
END LOOP;
```

Mark for Review
(1) Points

A FOR loop.

A WHILE loop.

A basic loop. (*)

An infinite loop.

A nested loop.

Incorrect Incorrect. Refer to Section 4.
2. For which one of these tasks should you use a PL/SQL loop? Mark for Review
(1) Points

Updating the salary of one employee.

Executing the same set of statements repeatedly until a condition becomes true. (*)

Deciding whether a value is within a range of numbers.

Making a decision based on whether a condition is true or not.

Incorrect Incorrect. Refer to Section 4.
3. What are the three kinds of loops in PL/SQL? Mark for Review
(1) Points

ascending, descending, unordered

infinite, finite, recursive

IF, CASE, LOOP

FOR, WHILE, basic (*)

Incorrect Incorrect. Refer to Section 4.
4. How many EXIT statements can be coded inside a basic loop? Mark for Review
(1) Points

None.

One only.

Two.

As many as you need, there is no limit. (*)

Correct Correct
5. Look at this code:

```
DECLARE
v_bool BOOLEAN := TRUE;
v_date DATE;
BEGIN
LOOP
EXIT WHEN v_bool;
SELECT SYSDATE INTO v_date FROM dual;
END LOOP;
END;
```

How many times will the SELECT statement execute?
Mark for Review
(1) Points

Once.

Twice.

Never (the SELECT will not execute at all) (*)

An infinite number of times because the EXIT condition will never be true

Incorrect Incorrect. Refer to Section 4.
6. Examine the following code:

```
DECLARE
v_count NUMBER := 0;
v_string VARCHAR2(20);
BEGIN
```

```
LOOP
v_string := v_string || 'x';
IF LENGTH(v_string) > 10 THEN
EXIT;
END IF;
v_count := v_count + 1;
END LOOP;
DBMS_OUTPUT.PUT_LINE(v_count);
END;
```

What will be displayed when this block is executed?
Mark for Review
(1) Points

9

10 (*)

11

xxxxxxxxxxx

Incorrect Incorrect. Refer to Section 4.

7. What will be displayed when this block is executed?

```
DECLARE
v_count NUMBER := 10;
v_result NUMBER;
BEGIN
LOOP
v_count := v_count - 1;
EXIT WHEN v_count < 5;
v_result := v_count * 2;
END LOOP;
DBMS_OUTPUT.PUT_LINE(v_result);
END;
```

Mark for Review
(1) Points

8

10 (*)

12

NULL

Incorrect Incorrect. Refer to Section 4.

8. You want to calculate and display the multiplication table for "sevens": 7x1=7, 7x2=14, 7x3=21 and so on. Which kind of PL/SQL construct is best for this?

PlSQL feedback midterm semister 1 part2

Mark for Review
(1) Points

A loop (*)

A CASE statement

IF ... END IF;

A Boolean variable.

Incorrect Incorrect. Refer to Section 4.

1. In a WHILE loop, the controlling condition is checked at the start of each iteration. True or False? Mark for Review
(1) Points

True (*)

False

Incorrect Incorrect. Refer to Section 4.

2. In a FOR loop, an explicitly declared counter is automatically incremented by 1 for each iteration of the loop. True or False? Mark for Review
(1) Points

True

False (*)

Incorrect Incorrect. Refer to Section 4.

3. Look at this code fragment:

```
FOR i IN 1 .. 3 LOOP
i := 4;
DBMS_OUTPUT.PUT_LINE('The counter is: ' || i);
END LOOP;
```

How many lines of output will be displayed?
Mark for Review
(1) Points

One

Three

Four

The block will fail because you cannot change the value of i inside the loop. (*)

Incorrect Incorrect. Refer to Section 4.

4. Which statement best describes when a FOR loop should be used? Mark for Review
(1) Points

When an EXIT WHEN statement must be coded.

When an implicitly declared counter must increase by 1 in each iteration of the loop. (*)

When we want to exit from the loop when a Boolean variable becomes FALSE.

When the statements inside the loop must execute at least once.

Incorrect Incorrect. Refer to Section 4.

5. You want a loop that counts backwards from 10 through 1. How do you code that? Mark for Review
(1) Points

FOR i IN 10 .. 1 LOOP

FOR i IN 1 .. 10 BY -1 LOOP

FOR i IN REVERSE 1 .. 10 LOOP (*)

FOR i IN REVERSE 10 .. 1 LOOP

Incorrect Incorrect. Refer to Section 4.

6. Look at the following code fragment:

```
i := 2;
WHILE i < 3 LOOP
i := 4;
DBMS_OUTPUT.PUT_LINE('The counter is: ' || i);
END LOOP;
```

How many lines of output will be displayed?
Mark for Review
(1) Points

No lines

One line (*)

Two lines

The block will fail because you cannot use DBMS_OUTPUT.PUT_LINE inside a loop.

Incorrect Incorrect. Refer to Section 4.

7. Look at the following block:

```
DECLARE
v_date DATE := SYSDATE;
BEGIN
WHILE v_date < LAST_DAY(v_date) LOOP
v_date := v_date + 1;
END LOOP;
DBMS_OUTPUT.PUT_LINE(v_date);
END;
```

If today's date is 17th April 2007, what will be displayed when this block executes?

Mark for Review

(1) Points

01-MAY-07

31-DEC-07

4/30/2007 (*)

4/17/2007

Correct Correct

8. You should use a WHILE loop when the number of iterations of the loop is known in advance. True or False? Mark for Review

(1) Points

True

False (*)

Correct Correct

1. Which one of these statements about using nested loops is true? Mark for Review

(1) Points

All the loops must be labelled

The outer loop must be labelled, but the inner loop need not be labelled

The outer loop must be labelled if you want to exit the outer loop from within the inner loop (*)

Both loops can have the same label

Correct Correct

2. When the following code is executed, how many lines of output will be displayed?

```
BEGIN
FOR i IN 1..5 LOOP
FOR j IN 1..8 LOOP
DBMS_OUTPUT.PUT_LINE(i || ',' || j);
END LOOP;
DBMS_OUTPUT.PUT_LINE(i);
END LOOP;
END;
```

Mark for Review
(1) Points

80

45 (*)

14

41

Correct Correct

3. What will be displayed when the following block is executed?:

```
DECLARE
x NUMBER(6) := 0 ;
BEGIN
FOR i IN 1..10 LOOP
FOR j IN 1..5 LOOP
x := x+1 ;
END LOOP;
END LOOP;
DBMS_OUTPUT.PUT_LINE(x);
END;
```

Mark for Review
(1) Points

5

10

15

50 (*)

Incorrect Incorrect. Refer to Section 4.

4. Look at the following code:

```
DECLARE
v_blue NUMBER(3) := 0;
v_red NUMBER(3) := 0;
BEGIN
<<blue>> LOOP
v_blue := v_blue + 1;
EXIT WHEN v_blue > 10;
<<red>> LOOP
v_red := v_red + 1;
EXIT WHEN v_red > 10;
-- Line A
END LOOP red;
END LOOP blue;
END;
```

What should you code at Line A to exit from the outer loop?

Mark for Review

(1) Points

EXIT;

EXIT red;

EXIT <<blue>>;

EXIT blue; (*)

Incorrect Incorrect. Refer to Section 4.

1. What is wrong with the following code?

```
DECLARE
CURSOR emp_curs IS SELECT last_name, salary FROM employees;
v_last_name employees.last_name%TYPE;
v_salary employees.salary%TYPE;
BEGIN
FETCH emp_curs INTO v_last_name, v_salary;
OPEN emp_curs;
FETCH emp_curs INTO v_last_name, v_salary;
CLOSE emp_curs;
END;
```

Mark for Review

(1) Points

When FETCHing more than one row, you MUST use a loop.

The cursor declaration does not include a WHERE condition.

The cursor declaration does not include an INTO clause.

The first row is FETCHed before the cursor is OPENed. (*)

Incorrect Incorrect. Refer to Section 5.

2. Which of the following best describes the difference between implicit and explicit cursors? Mark for Review
(1) Points

Implicit cursors are used for SELECT statements, while explicit cursors are used for DML statements.

Implicit cursor are named by the PL/SQL programmer, while explicit cursors are always named SQL.

Implicit cursors are defined automatically by Oracle, while explicit cursors must be declared by the PL/SQL programmer. (*)

Implicit cursors store rows on disk, while explicit cursors store rows in memory.

Correct Correct

3. There are 8 countries in REGION_ID 13 (Central America). What will happen when the following code is executed?

```
DECLARE
CURSOR country_curs IS SELECT country_name FROM wf_countries
WHERE region_id = 13;
v_country_name wf_countries.country_name%TYPE;
BEGIN
OPEN country_curs;
WHILE country_curs%FOUND
LOOP
FETCH country_curs INTO v_country_name;
DBMS_OUTPUT.PUT_LINE(v_country_name);
END LOOP;
CLOSE country_curs;
END;
```

Mark for Review
(1) Points

Eight rows will be fetched and displayed successfully.

The last seven rows will be fetched and displayed.

The block will execute, but no rows will be displayed. (*)

The block will fail because you can not use a WHILE loop with an explicit cursor.

None of the above.

Incorrect Incorrect. Refer to Section 5.

4. You execute the following code:

```
DECLARE
CURSOR emp_curs IS SELECT last_name FROM employees;
v_last_name employees.last_name%TYPE;
BEGIN
OPEN emp_curs;
LOOP -- Point A
FETCH emp_curs INTO v_last_name;
EXIT WHEN emp_curs%NOTFOUND;
DBMS_OUTPUT.PUT_LINE(v_last_name);
END LOOP;
CLOSE emp_curs;
END;
```

At Point A (after you have OPENed the cursor) another user updates an employee's last_name from 'Smith' to 'Jones' and immediately COMMITs.

When your block FETCHes this row, which value will be fetched and displayed?

Mark for Review

(1) Points

1

Smith (*)

Jones

Smith and Jones (the row will be fetched twice)

An INVALID_CURSOR exception will be raised when you try to FETCH the row.

Incorrect Incorrect. Refer to Section 5.

5. What is wrong with the following code?

```
DECLARE
CURSOR dept_curs IS SELECT department_name FROM departments;
v_dept_name departments.department_name%TYPE;
BEGIN
OPEN dept_curs;
LOOP
FETCH dept_curs INTO v_dept_name;
EXIT WHEN dept_curs%NOTFOUND;
DBMS_OUTPUT.PUT_LINE(v_dept_name);
CLOSE dept_curs;
```

```
END LOOP;
END;
```

Mark for Review
(1) Points

Nothing is wrong, all the rows will be FETCHed and displayed.

The OPEN statement should be inside the loop.

The EXIT WHEN ... statement should be coded outside the loop.

The CLOSE statement should be coded after END LOOP; (*)

The loop should be a WHILE loop, not a basic loop.

Correct Correct
6. When must you declare and use an explicit cursor? Mark for Review
(1) Points

You need to UPDATE more than one row in a table.

You want to use a MERGE statement.

You need to SELECT more than one row from a table. (*)

You want to be able to ROLLBACK a transaction if needed.

Correct Correct
7. Which one of the following statements is NOT true? Mark for Review
(1) Points

You can use ORDER BY when declaring an explicit cursor.

You can not use an INTO clause when declaring an explicit cursor.

An explicit cursor can select from only one table. No joins are allowed. (*)

An explicit cursor must be DECLAREd before it can be OPENed.

Correct Correct
8. You cannot OPEN or CLOSE an implicit cursor. Why not? Mark for Review
(1) Points

Because an implicit cursor is always called SQL.

Because an implicit cursor is OPENed and CLOSEd automatically by Oracle. (*)

Correct Correct
9. Examine the following code:

```
DECLARE
CURSOR dept_curs IS SELECT department_name FROM departments;
v_dept_name departments.department_name%TYPE;
BEGIN
OPEN dept_curs;
LOOP
FETCH dept_curs INTO v_dept_name;
DBMS_OUTPUT.PUT_LINE(v_dept_name);
EXIT WHEN dept_curs%NOTFOUND;
END LOOP;
CLOSE dept_curs;
END;
```

There are 10 rows in the DEPARTMENTS table. What will happen when this code is executed?
Mark for Review
(1) Points

10 rows will be displayed.

10 rows will be displayed, followed by a row of NULL values.

The last row will be displayed twice. (*)

A NO_DATA_FOUND exception will be raised.

The loop will execute for ever; the same 10 rows will be displayed over and over again.

Incorrect Incorrect. Refer to Section 5
10. You have declared a cursor EMP_CURSOR to select many rows from the EMPLOYEES table. The following five statements will be in the executable section:

A. FETCH emp_cursor INTO v_empno,v_last_name;
B. OPEN emp_cursor;
C. END LOOP;
D. CLOSE emp_cursor;
E. LOOP

In which order should you code these statements?
Mark for Review
(1) Points

B, E, A, C, D (*)

E, B, A, C, D

B, E, A, D, C

B, A, E, D, C

Incorrect Incorrect. Refer to Section 5.
11. One (and only one) employee has LAST_NAME = 'Grant'. You need to code:

```
SELECT ... FROM employees WHERE last_name = 'Grant';
```

Which type of cursor should you use, and why?

Mark for Review
(1) Points

An implicit cursor, because there is only one 'Grant'.

An implicit cursor, because SELECT is a SQL statement and implicit cursors are always called "SQL".

An explicit cursor, because there could be more than one 'Grant' in the future. (*)

An explicit cursor, because you can use an implicit cursor only for DML statements.

Correct Correct
2. Which one of the following explicit cursor declarations is NOT valid? Mark for Review
(1) Points

```
CURSOR country_curs IS
SELECT country_name, region_name
FROM wf_countries c, wf_world_regions r
WHERE c.region_id = r.region_id;
```

```
CURSOR country_curs IS
SELECT country_name INTO v_country_name
FROM wf_countries;
```

(*)

```
CURSOR country_curs IS
SELECT country_name
FROM wf_countries
ORDER BY population DESC;
```

```
        CURSOR country_curs IS
SELECT country_name
FROM wf_countries
WHERE region_id IN
(SELECT region_id FROM wf_world_regions
WHERE LOWER(region_name) LIKE '%asia%');
```

Incorrect Incorrect. Refer to Section 5.

1. Examine the following code:

```
DECLARE
CURSOR country_curs IS
SELECT country_id, country_name
FROM wf_countries
ORDER BY country_name;
v_country country_curs%ROWTYPE;
BEGIN
OPEN country_curs;
LOOP
FETCH country_curs INTO v_country;
EXIT WHEN country_curs%NOTFOUND;
------- Line A
END LOOP;
CLOSE country_curs;
END;
```

You want to display the id and name of each FETCHed country. What would you code at Line A?

Mark for Review

(1) Points

```
        DBMS_OUTPUT.PUT_LINE(country_id || ' ' || country_name);
```

```
        DBMS_OUTPUT.PUT_LINE(v_country(country_id) || ' ' ||
v_country(country_name));
```

```
        DBMS_OUTPUT.PUT_LINE(country_curs.country_id || ' ' ||
country_curs.country_name);
```

```
        DBMS_OUTPUT.PUT_LINE(v_country.country_id || ' ' || v_country.country_name);
(*)
```

Incorrect Incorrect. Refer to Section 5.

2. How must you reference one field which is part of a PL/SQL record? Mark for Review

(1) Points

```
        field_name.record_name
```

record_name.field_name (*)

record_name(field_name)

field_name OF record_name

It cannot be done.

Incorrect Incorrect. Refer to Section 5.

3. You have declared the following cursor:

```
CURSOR country_curs IS
SELECT * FROM wf_countries
ORDER BY country_name;
```

There are over 200 rows in the WF_COUNTRIES table, but you want to fetch and display only the first 25 rows.

How would you exit from the FETCH loop?

Mark for Review

(1) Points

EXIT WHEN country_curs%FOUND(25);

EXIT WHEN country_curs%ROWCOUNT > 25; (*)

EXIT WHEN ROWCOUNT > 25;

WHEN country_curs > 25 THEN EXIT; END IF;

Incorrect Incorrect. Refer to Section 5.

4. Look at these declarations:

```
DECLARE
CURSOR dept_loc_cursor IS
SELECT department_id, department_name, location_name
FROM departments d, locations l
WHERE d.location_id = l.location_id;
v_dept_loc dept_loc_cursor%ROWTYPE;
```

How many fields does V_DEPT_LOC contain?

Mark for Review

(1) Points

Two, because the cursor joins two tables

Four

Three (*)

None

Correct Correct
5. Look at the following code:

```
DECLARE
CURSOR emp_cursor IS
SELECT employee_id, last_name, salary FROM employees;
v_empcurs emp_cursor%ROWTYPE;
```

What is the data type of V_EMPCURS?
Mark for Review
(1) Points

Scalar

Record (*)

Cursor

Row

Incorrect Incorrect. Refer to Section 5.
6. Which of the following explicit cursor attributes evaluates to TRUE if the most recent FETCH returns a row? Mark for Review
(1) Points

%ISOPEN

%NOTFOUND

%FOUND (*)

%ROWCOUNT

Incorrect Incorrect. Refer to Section 5.
7. You can reference explicit cursor attributes directly in a SQL statement. True or False? Mark for Review
(1) Points

True

False (*)

Incorrect Incorrect. Refer to Section 5.
1. What is the DISadvantage of using a cursor FOR loop with a subquery? Mark for Review
(1) Points

You cannot reference cursor attributes such as %NOTFOUND. (*)

The execution speed is slower.

You cannot declare the cursor in the declaration section.

You cannot use the cursor to join two or more tables.

There are no disadvantages.

Incorrect Incorrect. Refer to Section 5
2. You have declared a cursor as follows:

```
CURSOR loc_curs IS SELECT * FROM locations;
```

How should you code a FOR loop to use this cursor?
Mark for Review
(1) Points

FOR loc_rec IN 1 .. loc_curs%ROWCOUNT LOOP ...

WHILE loc_rec IN loc_curs LOOP ...

FOR loc_curs IN loc_rec LOOP ...

IF loc_rec IN loc_curs LOOP ...

FOR loc_rec IN loc_curs LOOP ... (*)

3. What is wrong with the following piece of code?

```
BEGIN
FOR emp_record IN emp_cursor LOOP
DBMS_OUTPUT.PUT_LINE(emp_record.last_name);
END LOOP;
IF emp_record.last_name = 'Patel' THEN ...
```

Mark for Review
(1) Points

EMP_RECORD has not been explicitly declared.

The cursor has not been OPENed.

You cannot reference EMP_RECORD outside the loop. (*)

It should read: DBMS_OUTPUT.PUT_LINE(emp_cursor.last_name);

Nothing is wrong, the code will execute correctly.

Incorrect Incorrect. Refer to Section 5

4. Which of the following is a benefit of using a cursor FOR loop? Mark for Review
(1) Points

The exception handling is done automatically. .

The OPEN, CLOSE, FETCH and EXIT from the loop are done automatically. (*)

You can OPEN the same cursor twice at the same time.

Because there is less code, the loop executes faster.

%ROWCOUNT increments automatically each time a row is FETCHed.

Incorrect Incorrect. Refer to Section 5

5. Which one of the following is a valid cursor FOR loop with a subquery? Mark for Review
(1) Points

FOR emp_rec IN (SELECT last_name || first_name FROM employees) LOOP ...

FOR emp_rec IN (SELECT UPPER(last_name) FROM employees) LOOP ...

FOR emp_rec IN SELECT last_name, salary*12 "ANNSAL" FROM employees LOOP ...

FOR emp_rec IN (SELECT last_name, salary*12 "ANNSAL" FROM employees) LOOP ... (*)

None of the above.

Incorrect Incorrect. Refer to Section 5

6. Look at the following code:

```
DECLARE
CURSOR emp_cursor IS SELECT * FROM employees;
BEGIN
FOR emp_record IN emp_cursor LOOP
```

```
DBMS_OUTPUT.PUT_LINE( --Point A -- );
END LOOP;
END;
```

To display the salary of an employee, what code should you write at Point A?
Mark for Review
(1) Points

emp_record.salary (*)

emp_cursor.salary

employees.salary

emp_record.employees.salary

TO_CHAR(salary)

Incorrect — Incorrect. Refer to Section 5

1. The following cursor has been declared:

```
CURSOR emp_curs
(p_dept_id employees.department_id%TYPE,
p_job_id employees.job_id%TYPE) IS
SELECT * FROM employees
WHERE department_id = p_dept_id
AND job_id = p_job_id;
```

Which of the following will correctly open the cursor?
Mark for Review
(1) Points

OPEN emp_curs(20);

FOR emp_rec IN emp_curs(20) LOOP ...

OPEN emp_curs('IT_PROG', 20);

FOR emp_rec IN emp_curs(20,'IT_PROG') LOOP ... (*)

FOR emp_rec IN emp_curs(p_dept_id p_job_id) LOOP ...

Incorrect — Incorrect. Refer to Section 5

2. Look at the following code:

```
DECLARE
CURSOR emp_curs (p_dept_id employees.department_id%TYPE) IS
SELECT * FROM employees
WHERE department_id = p_dept_id;
```

PlSQL feedback midterm semister 1 part2

```
v_emp_rec emp_curs%ROWTYPE;
v_deptid NUMBER(4) := 50;
BEGIN
OPEN emp_curs( -- Point A --);
....
```

You want to open the cursor, passing value 50 to the parameter. Which of the following are correct at Point A?
Mark for Review
(1) Points

50

v_deptid

100 / 2

All of the above. (*)

Incorrect Incorrect. Refer to Section 5.

3. Using parameters with a cursor, you can open and close the cursor several times in a block, returning a different active set each time. True or False? Mark for Review
(1) Points

True (*)

False

Incorrect Incorrect. Refer to Section 5.

4. You want to use explicit cursors to fetch and display all the countries in a specific region. There are 19 rows in the WF_WORLD_REGIONS table. You want to use a different region each time the cursor is opened. How many cursors should you declare? Mark for Review
(1) Points

19 cursors, all in the same PL/SQL block.

19 cursors in 19 PL/SQL blocks (one in each block).

20 cursors, in case an extra row is inserted into WF_WORLD_REGIONS later.

One cursor with a parameter in the WHERE clause. (*)

None of the above.

Incorrect Incorrect. Refer to Section 5.

5. What is wrong with the following cursor declaration?

```
CURSOR dept_curs (p_loc_id NUMBER(4)) IS
SELECT * FROM departments
WHERE location_id = p_loc_id;
```

Mark for Review
(1) Points

You cannot reference a cursor parameter in a WHERE clause.

The parameter should be coded as: (p_loc_id NUMBER) (*)

The parameter should be coded as: (p_loc_id IN NUMBER)

Nothing is wrong, the cursor declaration is correct.

Incorrect Incorrect. Refer to Section 5.

1. What is the difference between the following two blocks of code?

```
--Block A
DECLARE
    CURSOR emp_cursor IS
    SELECT employee_id, last_name
    FROM employees
    WHERE department_id = 80
    FOR UPDATE OF salary;

--Block B
DECLARE
    CURSOR emp_cursor IS
    SELECT employee_id, last_name
    FROM employees
    WHERE department_id = 80
    FOR UPDATE OF salary
    NOWAIT;
```

Mark for Review
(1) Points

There is no difference; the programs behave exactly the same way.

In Block A, the program waits indefinitely until the rows are available. In Block B, the program returns control immediately so that it can do other work. (*)

In Block A, the program waits indefinitely until the rows are available. In Block B, control is returned to your program after 5 seconds so that it can do other work.

Correct Correct

2. You have declared a cursor as SELECT .... FOR UPDATE; You have OPENed the cursor and locked the FETCHed rows. When are these row locks released? Mark for Review

(1) Points

When an UPDATE ... WHERE CURRENT OF cursor_name; is executed.

When you CLOSE the cursor.

When your block finishes executing.

When you explicitly COMMIT or ROLLBACK your transaction. (*)

When another user tries to SELECT the rows.

Incorrect Incorrect. Refer to Section 5.

3. You want to fetch rows from the EMPLOYEES table. You want to lock the fetched rows, to prevent other users from updating them.
You declare the following cursor:

```
CURSOR emp_curs IS
SELECT employee_id, last_name, salary
FROM employees
-- Line A -- ;
```

What should you code at Line A?

Mark for Review

(1) Points

FOR LOCK

FOR UPDATE OF employees

FOR UPDATE (*)

FOR UPDATE (employees)

Correct Correct

4. You have declared the following cursor:

```
CURSOR country_curs IS
SELECT country_id, country_name
FROM wf_countries
FOR UPDATE WAIT 10;
```

Another user updates a row in WF_COUNTRIES but does not COMMIT the update. What will happen when you OPEN country_curs; ?

Mark for Review

(1) Points

A LOCKED_ROWS exception is raised immediately.

The other user's transaction is automatically rolled back.

Your session waits indefinitely until the other user COMMITs.

Your session waits for 10 seconds, and then returns control to your block so that it can continue to execute. (*)

Your block fails because you should have coded: FOR UPDATE WAIT (10);

Correct Correct
5. Why can we NOT code:
INSERT INTO table-name
WHERE CURRENT OF cursor_name;
Mark for Review
(1) Points

Because the syntax is wrong. An INSERT statement must have a VALUES (....) clause.

Because the syntax is wrong. It should be: INSERT INTO cursor-name .... WHERE CURRENT OF table-name;

Because WHERE CURRENT OF ... modifies the most recently FETCHed row, and you cannot FETCH a row that is not in the table yet. (*)

Because another user has locked the rows and not committed.

Nothing is wrong; we CAN code: INSERT .... WHERE CURRENT OF ... ;

Incorrect Incorrect. Refer to Section 5.
6. When can we use the WHERE CURRENT OF clause? Mark for Review
(1) Points

Only with an UPDATE, not with a DELETE.

Only with a DELETE, not with an UPDATE.

When the cursor is declared as SELECT ... FOR UPDATE ...; (*)

When the cursor is based on a single table (not on a join).

When the cursor has not been OPENed.

Correct Correct
7. You declare a cursor as a join of two tables:

```
CURSOR emp_dept_curs IS
SELECT last_name, salary, department_name
FROM employees e, departments d
WHERE e.department_id = d.department_id
-- Point A -- ;
```

You want to lock fetched rows from EMPLOYEES, but NOT lock fetched rows from DEPARTMENTS.

Which of the following is correct at Point A?
Mark for Review
(1) Points

FOR UPDATE

FOR UPDATE of salary (*)

FOR UPDATE OF employees

FOR UPDATE (last_name)

Incorrect Incorrect. Refer to Section 5.
1. Which of the following is NOT allowed when using multiple cursors with parameters? Mark for Review
(1) Points

You cannot use cursor FOR loops.

You cannot declare the cursors FOR UPDATE.

You cannot declare a cursor based on a join.

You cannot OPEN more than one cursor at the same time.

None of the above, they are all allowed. (*)

Incorrect Incorrect. Refer to Section 5.
2. Which of the following is a good reason to use two cursors in a single PL/SQL block? Mark for Review
(1) Points

To allow one cursor to be opened twice at the same time.

When two tables are related to each other (often by a foreign key) and we want to produce a multilevel report using data from both tables. (*)

To allow rows to be locked as they are FETCHed.

To speed up the execution of the PL/SQL block.

It is the only way to declare a cursor with a parameter.

Incorrect Incorrect. Refer to Section 5.

3. Assume your schema contains 25 tables. How many explicit cursors can you declare and use within a single PL/SQL block? Mark for Review
(1) Points

Only one.

As many as you need - there is no limit. (*)

A maximum of three.

As many as you need, but only one of them can be open at any time.

A maximum of 25 (one for each table in your schema).

Incorrect Incorrect. Refer to Section 5.

4. Assume that table BIGDEPTS contains 100 rows, and table BIGEMPS contains 1000 rows, with 10 employees in each department. Consider the following code:

```
DECLARE
CURSOR bigdept_cur IS
SELECT * FROM bigdepts;
CURSOR bigemp_cur IS
SELECT * FROM bigemps;
BEGIN
FOR dept_rec IN bigdept_cur LOOP
DBMS_OUTPUT.PUT_LINE
(dept_rec.department_name);
FOR emp_rec IN bigemp_cur LOOP
IF emp_rec.department_id=dept_rec.department_id
THEN DBMS_OUTPUT.PUT_LINE
(emp_rec.last_name);
END IF;
END LOOP;
END LOOP;
END;
```

Why is this code inefficient?
Mark for Review
(1) Points

It locks both tables unnecessarily.

It is using two cursors when one cursor is enough.

It is doing a Cartesian Product, joining every employee with every department and displaying 1100 lines of output.

It reads 1000 employee rows every time BIGEMP_CUR is OPENed, and then ignores 990 of them. (*)

It is using cursor FOR loops, which are less efficient than OPENing and CLOSEing the cursors explicitly.

Incorrect Incorrect. Refer to Section 5.

5. You want to produce a report which displays each department and (immediately after each department) a list of employees who work in that department. You declare a DEPARTMENTS cursor as:

```
CURSOR dept_curs IS
SELECT * FROM departments
ORDER BY department_id;
```

How could you declare the EMPLOYEES cursor? (Choose two).

Mark for Review
(1) Points

(Choose all correct answers)

CURSOR emp_curs IS SELECT * FROM employees;

CURSOR emp_curs (p_dept_id NUMBER) IS SELECT * FROM employees WHERE department_id = p_dept_id; (*)

CURSOR emp_curs IS SELECT * FROM employees ORDER BY department_id;

CURSOR emp_curs (p_dept_id departments.department_id%TYPE) IS SELECT * FROM employees WHERE department_id = p_dept_id; (*)

CURSOR emp_curs IS SELECT * FROM employees WHERE department_id = departments.department_id;

Incorrect Incorrect. Refer to Section 5.

6. Examine the following code:

```
DECLARE
CURSOR region_cur IS
SELECT * FROM wf_world_regions;
v_region_rec region_cur%ROWTYPE;
CURSOR country_cur (p_region_id NUMBER) IS
SELECT * FROM wf_countries
WHERE region_id = p_region_id;
```

```
v_country_rec country_cur%ROWTYPE;
BEGIN
OPEN region_cur;
LOOP
FETCH region_cur INTO v_region_rec;
EXIT WHEN region_cur%NOTFOUND;
DBMS_OUTPUT.PUT_LINE
(v_region_rec.region_name);
-- Line A --
LOOP
FETCH country_cur INTO v_country_rec;
EXIT WHEN country_cur%NOTFOUND;
.......
```

What would you code at Line A?
Mark for Review
(1) Points

OPEN country_cur (p_region_id);

OPEN country_cur (wf_world_regions.region_id);

OPEN country_cur (v_region_rec.region_id); (*)

OPEN country_cur (region_cur.region_id);

OPEN country_cur;

Correct Correct

1. Errors are handled in the Exception part of the PL/SQL block. True or False?
Mark for Review
(1) Points

True (*)

False

Incorrect Incorrect. Refer to Section 1.

2. In which part of the PL/SQL block are declarations of variables defined?
Mark for Review
(1) Points

Executable

Exception

Declarative (*)

Definition

Incorrect Incorrect. Refer to Section 1.

3. Which of the following tools can NOT be used to develop and test PL/SQL code? Mark for Review
(1) Points

Oracle Jdeveloper

Oracle Application Express

Oracle JSQL (*)

Oracle iSQL*Plus

Incorrect Incorrect. Refer to Section 1.

4. Which component of Oracle Application Express is used to enter and run SQL statements and PL/SQL blocks? Mark for Review
(1) Points

Application Builder

SQL Workshop (*)

Utilities

Object Browser

Incorrect Incorrect. Refer to Section 1.

5. Which PL/SQL block type must return a value? Mark for Review
(1) Points

Anonymous

Function (*)

Procedure

Correct Correct

6. Given below are the parts of a PL/SQL block:

1. END;
2. EXCEPTION
3. DECLARE
4. BEGIN

Arrange the parts in order.
Mark for Review
(1) Points

2,1,4,3

3,4,2,1 (*)

3,2,4,1

4,3,2,1

Incorrect Incorrect. Refer to Section 1.

7. What is the purpose of using DBMS_OUTPUT.PUT_LINE in a PL/SQL block? Mark for Review
(1) Points

To perform conditional tests

To allow a set of statements to be executed repeatedly

To display results to check if our code is working correctly (*)

To store new rows in the database

Incorrect Incorrect. Refer to Section 1.

8. Which of the following can you use PL/SQL to do? Mark for Review
(1) Points

Update data (DML)

Develop Web applications using the Web Application Toolkit

Manage database security

Create customized reports

All of the above (*)

Incorrect Incorrect. Refer to Section 1.

9. PL/SQL can be used not only with an Oracle database, but also with any kind of relational database. True or False? Mark for Review (1) Points

True

False (*)

Correct Correct

10. The fact that PL/SQL is portable is a good thing because: Mark for Review (1) Points

Exceptions can be ported to different operating systems

Blocks can be sent to the operating system.

PL/SQL code can be developed on one platform and deployed on another (*)

PL/SQL code can be run on any operating system without a database

Correct Correct

11. PL/SQL extends SQL by including all of the following except: Mark for Review (1) Points

variables

conditional statements

reusable program units

constants

nonprocedural constructs (*)

Incorrect Incorrect. Refer to Section 1.

12. Which of the following statements about PL/SQL and SQL is true? Mark for Review
(1) Points

PL/SQL and SQL are both ANSI-compliant.

PL/SQL and SQL can be used with many types of databases, including Oracle.

PL/SQL and SQL are both Oracle proprietary programming languages.

PL/SQL allows basic program logic and control flow to be combined with SQL statements. (*)

Incorrect Incorrect. Refer to Section 1.

13. A program which specifies a list of operations to be performed sequentially to achieve the desired result can be called: Mark for Review
(1) Points

declarative

nondeclarative

procedural (*)

low level

Incorrect Incorrect. Refer to Section 1.

Section 2

14. A variable must have a value if NOT NULL is specified. True or False? Mark for Review
(1) Points

True (*)

False

Incorrect Incorrect. Refer to Section 2.

15. Which of the following declarations is invalid? Mark for Review
(1) Points

v_count PLS_INTEGER:=0;

college_name VARCHAR2(20):='Harvard';

v_pages CONSTANT NUMBER; (*)

v_start_date DATE := sysdate+1;

Correct Correct

16. Which of the following should NOT be used as the name of a variable? Mark for Review
(1) Points

A table name.

A table column name. (*)

The database name.

Correct Correct

17. When nested blocks are used, which blocks can or must be labeled? Mark for Review
(1) Points

The inner block must be labeled, the outer block can be labeled.

Both blocks must be labeled

Nested blocks cannot be labeled

The outer block must be labeled if it is to be referred to in the inner block. (*)

Incorrect Incorrect. Refer to Section 2.

18. When an exception occurs within a PL/SQL block, the remaining statements in the executable section of the block are skipped. True or False? Mark for Review
(1) Points

True (*)

False

Incorrect Incorrect. Refer to Section 2.

19. Examine the following code. At Line A, we want to assign a value of 22 to the outer block's variable v_myvar. What code should we write at Line A?

```
<<outer_block>>
DECLARE
    v_myvar NUMBER;
BEGIN
    <<inner_block>>
    DECLARE
       v_myvar NUMBER := 15;
    BEGIN
       -- Line A
    END;
END;
```

Mark for Review
(1) Points

outer_block.v_myvar := 22; (*)

v_myvar := 22;

<<outer_block>>.v_myvar := 22;

v_myvar(outer_block) := 22;

We cannot reference the outer block's variable because both variables have the same name

Incorrect Incorrect. Refer to Section 2.

PlSQL feedback midterm semister 1 part2

20. Examine the following code. Line A causes an exception. What will be displayed when the block is executed?

```
DECLARE
    var_a NUMBER := 6;
    var_b DATE;
BEGIN
    var_a := var_a * 2;
    var_b := '28 December 2006'; -- Line A
    var_a := var_a * 2;
EXCEPTION
    WHEN OTHERS THEN
       DBMS_OUTPUT.PUT_LINE(var_a);
END;
```

Mark for Review
(1) Points

12 (*)

24

6

Nothing will be displayed

Incorrect Incorrect. Refer to Section 2.

21. What will be displayed when the following code is executed?

```
DECLARE
    varA NUMBER := 12;
BEGIN
    DECLARE
       varB NUMBER := 8;
    BEGIN
       varA := varA + varB;
    END;
    DBMS_OUTPUT.PUT_LINE(varB);
END;
```

Mark for Review
(1) Points

8

12

Nothing, the block will fail with an error (*)

20

VarB

Incorrect Incorrect. Refer to Section 2.

22. Which of the following are valid assignment statements? (Choose two.) Mark for Review
(1) Points

(Choose all correct answers)

v_string = 'Hello';

v_string := Hello;

v_number := 17 + 34; (*)

v_string := 'Hello'; (*)

v_date := 28-DEC-06;

Incorrect Incorrect. Refer to Section 2.

23. Examine the following code. What is the final value of V_MYBOOL ?

```
DECLARE
    v_mynumber NUMBER;
    v_mybool BOOLEAN ;
BEGIN
    v_mynumber := 6;
    v_mybool := (v_mynumber BETWEEN 10 AND 20);
    v_mybool := NOT (v_mybool);
END;
```

Mark for Review
(1) Points

True (*)

False

Incorrect Incorrect. Refer to Section 2.

24. Examine the following code:

```
1 DECLARE
2 x NUMBER;
3 BEGIN
4 x:= '300';
5 END;
```

After line 4, what is the value of x?
Mark for Review
(1) Points

'300'

300 (*)

NULL

Correct Correct

25. The implicit data type conversion at Point A may not work correctly. Why not?

```
DECLARE
    v_mydate DATE;
BEGIN
    V_MYDATE := '29-Feb-04'; -- Point A
END;
```

Mark for Review
(1) Points

There are only 28 days in February

Oracle cannot implicitly convert a character string to a date, even if the string contains a valid date value

If the database language is not English, 'Feb' has no meaning. (*)

V_MYDATE has been entered in uppercase

Incorrect Incorrect. Refer to Section 2.

26. PL/SQL can convert a VARCHAR2 value containing alphabetic characters to a NUMBER value. True or False? Mark for Review
(1) Points

True

False (*)

Correct Correct

27. The DECODE function is available in PL/SQL procedural

statements. True or False? Mark for Review
(1) Points

True

False (*)

Incorrect Incorrect. Refer to Section 2.

28. What is wrong with this assignment statement?

myvar := 'To be or not to be';
'That is the question';
Mark for Review
(1) Points

An assignment statement must be a single line of code

Nothing is wrong, the statement is fine

An assignment statement must have a single semicolon at the end (*)

"myvar" is not a valid name for a variable

Character literals should not be enclosed in quotes

Correct Correct

29. Single row character functions are valid SQL functions in PL/SQL. True or False? Mark for Review
(1) Points

True (*)

False

Incorrect Incorrect. Refer to Section 2.

30. Which of the following are PL/SQL lexical units? (Choose two.) Mark for Review
(1) Points

(Choose all correct answers)

PlSQL feedback midterm semister 1 part2

Identifiers (*)

Table Columns

Reserved Words (*)

Anonymous Blocks

SQL Workshop

Incorrect Incorrect. Refer to Section 2.

31. Valid identifiers begin with a Mark for Review
(1) Points

Number

Letter (*)

Special character

Incorrect Incorrect. Refer to Section 2.

32. Which of the following are valid identifiers? (Choose two.) Mark for Review
(1) Points

(Choose all correct answers)

Full Name

students_street_address (*)

v_code (*)

#hours

completion_%

Incorrect Incorrect. Refer to Section 2.

33. Which statement most closely describes "data type"? Mark for Review
(1) Points

It is the value of a variable.

It specifies a storage format, constraints, and a valid range of values for a variable. (*)

It allows different kinds of data to be stored in a single variable.

It is used to test if errors have occurred.

Correct Correct

34. _____ are meant to store large amounts of data. Mark for Review (1) Points

Variables

Scalar data types

LOBs (*)

Incorrect Incorrect. Refer to Section 2.

35. A movie is an example of which category of data type? Mark for Review (1) Points

Scalar

Composite

Reference

LOB (*)

Incorrect Incorrect. Refer to Section 2.

36. Assignment statements can continue over several lines in PL/SQL. True or False? Mark for Review (1) Points

True (*)

False

Correct Correct

37. Variables can be assigned a value in both the Executable and Declaration sections of a PL/SQL program. True or False? Mark for Review (1) Points

True (*)

False

Incorrect Incorrect. Refer to Section 2.

38. When a variable is defined using the CONSTANT keyword, the value of the variable cannot change. True or False? Mark for Review (1) Points

True (*)

False

Correct Correct

39. Identify which of the following assignment statements are valid. (Choose three.) Mark for Review (1) Points

(Choose all correct answers)

v_last_name := Chandra;

v_blackout_date := '31-DEC-2006'; (*)

v_population := 333444; (*)

v_music_type := 'ROCK'; (*)

Incorrect Incorrect. Refer to Section 2.

PlSQL feedback midterm semister 1 part2

40. When a variable is defined using the NOT NULL keywords, the variable must contain a value. True or False? Mark for Review
(1) Points

True (*)

False

Correct Correct

41. Which of the following best describes a database transaction? Mark for Review
(1) Points

All the DML statements in a single PL/SQL block

A related set of SQL DML statements which must be executed either completely or not at all (*)

A single SQL statement that updates multiple rows of a table

A SELECT statement based on a join of two or more database tables

Correct Correct

42. The following anonymous block of code is run:

```
BEGIN
    INSERT INTO countries (id, name)
    VALUES ('XA', 'Xanadu');
    SAVEPOINT XA;
    INSERT INTO countries (id, name)
    VALUES ('NV','Neverland');
    COMMIT;
    ROLLBACK TO XA;
END;
```

What happens when the block of code finishes?
Mark for Review
(1) Points

No data is inserted and no errors occur.

No data is inserted and an error occurs

Two rows are inserted and no errors occur.

Two rows are inserted and an error occurs. (*)

Incorrect Incorrect. Refer to Section 3.

43. Which of the following is NOT a good guideline for retrieving data in PL/SQL? Mark for Review
(1) Points

Declare the receiving variables using %TYPE

The WHERE clause is optional in nearly all cases. (*)

Specify the same number of variables in the INTO clause as database columns in the SELECT clause.

THE SELECT statement should fetch exactly one row.

Incorrect Incorrect. Refer to Section 3.

44. Given this first section of code:

```
DECLARE
    v_result employees.salary%TYPE;
BEGIN
```

Which statement will always return exactly one value?
Mark for Review
(1) Points

```
SELECT salary
INTO v_result
FROM employees;
```

```
SELECT salary
INTO v_result
FROM employees
WHERE last_name ='Smith';
```

```
SELECT salary
INTO v_result
FROM employees
WHERE department_id = 80;
```

```
SELECT SUM(salary)
INTO v_result
FROM employees;
```

(*)

Incorrect Incorrect. Refer to Section 3.

45. Which one of these SQL statements can be directly included in a PL/SQL executable block? Mark for Review
(1) Points

```
SELECT last_name FROM employees
WHERE employee_id=100;
```

```
DESCRIBE employees;
```

```
UPDATE employees
SET last_name='Smith';
```

(*)

```
DROP TABLE employees;
```

Correct Correct

46. A variable is declared as:

```
DECLARE
    v_holdit employees.last_name%TYPE;
BEGIN ...
```

Which of the following is a correct use of the INTO clause?
Mark for Review
(1) Points

```
SELECT *
INTO v_holdit
FROM employees;
```

```
SELECT last_name
INTO v_holdit
FROM employees;
```

```
SELECT last_name
INTO v_holdit
FROM employees
WHERE employee_id=100;
```

(*)

```
        SELECT salary
INTO v_holdit
FROM employees
WHERE employee_id=100;
```

Incorrect Incorrect. Refer to Section 3.

47. Which one of these SQL statements can be directly included in a PL/SQL executable block? Mark for Review
(1) Points

```
        DELETE FROM employees
WHERE department_id=60;
```

(*)

```
        SELECT salary FROM employees
WHERE department_id=60;
```

```
        CREATE TABLE new_emps (last_name VARCHAR2(10), first_name VARCHAR2(10));
```

```
        DROP TABLE locations;
```

Incorrect Incorrect. Refer to Section 3.

48. You declare an implicit cursor in the DECLARE section of a PL/SQL block. True or False? Mark for Review
(1) Points

True

False (*)

Correct Correct

49. Which SQL statement can NOT use an implicit cursor? Mark for Review
(1) Points

A DELETE statement

An UPDATE statement

A SELECT statement that returns multiple rows (*)

A SELECT statement that returns one row

Correct Correct

50. A PL/SQL block includes the following statement:

```
SELECT last_name INTO v_last_name
FROM employees
WHERE employee_id=100;
```

What is the value of SQL%ISOPEN immediately after the SELECT statement is executed?
Mark for Review
(1) Points

True

False (*)

Null

Error. That attribute does not apply for implicit cursors.

Incorrect Incorrect. Refer to Section 3.

1. Comparing PL/SQL with other languages such as C and Java, which of the following statements is true? Mark for Review
(1) Points

PL/SQL is harder to learn

PL/SQL is easier to learn and more efficient (*)

PL/SQL is easier to learn but less efficient

PL/SQL is easier to learn and does not require an Oracle database or tool

Correct Correct

2. Using Oracle Application Express, you can create Web applications that include PL/SQL. True or False? Mark for Review
(1) Points

True (*)

False

Incorrect Incorrect. Refer to Section 1.

3. Which of the following can you use PL/SQL to do? Mark for Review
(1) Points

Update data (DML)

Develop Web applications using the Web Application Toolkit

Manage database security

Create customized reports

All of the above (*)

Incorrect Incorrect. Refer to Section 1.

4. A program which specifies a list of operations to be performed sequentially to achieve the desired result can be called: Mark for Review
(1) Points

declarative

nondeclarative

procedural (*)

low level

Correct Correct

5. The P in PL/SQL stands for: Mark for Review
(1) Points

Processing

Procedural (*)

Primary

Proprietary

Correct Correct

6. SQL is a common access language for many types of databases, including Oracle. True or False? Mark for Review
(1) Points

True (*)

False

Incorrect Incorrect, Refer to Section 1.

7. Every PL/SQL anonymous block must start with the keyword DECLARE. True or False? Mark for Review
(1) Points

True

False (*)

Incorrect Incorrect. Refer to Section 1.

8. In which part of the PL/SQL block are declarations of variables defined? Mark for Review
(1) Points

Executable

Exception

Declarative (*)

Definition

Correct Correct

9. Which statements are optional in a PL/SQL block? (Choose two.) Mark for Review
(1) Points

(Choose all correct answers)

DECLARE (*)

BEGIN

EXCEPTION (*)

END;

Correct Correct

10. Which lines of code will correctly display the message "The cat sat on the mat"? (Choose two.) Mark for Review
(1) Points

(Choose all correct answers)

DBMS_OUTPUT.PUT_LINE('The cat sat on the mat'); (*)

DBMS_OUTPUT.PUT_LINE(The cat sat on the mat);

DBMS_OUTPUT.PUT_LINE('The cat' || 'sat on the mat');

DBMS_OUTPUT.PUT_LINE('The cat sat ' || 'on the mat'); (*)

Incorrect Incorrect. Refer to Section 1.

11. Which of the following tools can NOT be used to develop and test PL/SQL code? Mark for Review
(1) Points

Oracle Jdeveloper

Oracle Application Express

Oracle JSQL (*)

Oracle iSQL*Plus

Incorrect Incorrect. Refer to Section 1.

12. What is the purpose of using DBMS_OUTPUT.PUT_LINE in a PL/SQL block? Mark for Review
(1) Points

To perform conditional tests

To allow a set of statements to be executed repeatedly

To display results to check if our code is working correctly (*)

To store new rows in the database

Correct Correct

13. Which PL/SQL block type must return a value? Mark for Review
(1) Points

Anonymous

Function (*)

Procedure

Incorrect Incorrect. Refer to Section 1.

Section 2

14. 1. Null
2. False
3. True
4. 0

Which of the above can be assigned to a Boolean variable?
Mark for Review
(1) Points

2 and 3

2, 3 and 4

1, 2 and 3 (*)

1, 2, 3 and 4

Correct Correct

15. You need to declare a variable to hold a value which has been read from the SALARY column of the EMPLOYEES table. Which of the following is an advantage of declaring the variable as: employees.salary%TYPE ? Mark for Review
(1) Points

It is shorter than coding NUMBER(8,2)

If the SALARY column is ALTERed later, the PL/SQL code need not be changed. (*)

It executes much faster than using NUMBER(8,2)

It allows the software to perform implicit data type conversions.

Incorrect Incorrect. Refer to Section 2.

16. Which of the following should NOT be used as the name of a variable? Mark for Review
(1) Points

A table name.

A table column name. (*)

The database name.

Correct Correct

17. Delimiters are _____ that have special meaning to the Oracle database. Mark for Review
(1) Points

identifiers

variables

symbols (*)

Correct Correct

18. Which of the following are valid identifiers? (Choose two.)
Mark for Review
(1) Points

(Choose all correct answers)

Full Name

students_street_address (*)

v_code (*)

#hours

completion_%

Correct Correct

19. Which statements about lexical units are true? (Choose two.)
Mark for Review
(1) Points

(Choose all correct answers)

They are named objects stored in the database

They are the building blocks of every PL/SQL program (*)

They are optional but can make a PL/SQL block execute faster

They are sequences of characters including letters, digits, tabs, returns and symbols (*)

Correct Correct

20. What will be displayed when the following code is executed?

```
DECLARE
    varA NUMBER := 12;
BEGIN
    DECLARE
       varB NUMBER := 8;
```

```
    BEGIN
       varA := varA + varB;
    END;
    DBMS_OUTPUT.PUT_LINE(varB);
END;
```

Mark for Review
(1) Points

8

12

Nothing, the block will fail with an error (*)

20

VarB

Correct Correct

21. When an exception occurs within a PL/SQL block, the remaining statements in the executable section of the block are skipped. True or False? Mark for Review
(1) Points

True (*)

False

Correct Correct

22. When nested blocks are used, which blocks can or must be labeled? Mark for Review
(1) Points

The inner block must be labeled, the outer block can be labeled.

Both blocks must be labeled

Nested blocks cannot be labeled

The outer block must be labeled if it is to be referred to in the inner block. (*)

Correct Correct

23. In the following code, Line A causes an exception. What value will be displayed when the code is executed?

```
DECLARE
    outer_var VARCHAR2(50) := 'My';
BEGIN
    outer_var := outer_var || ' name';
    DECLARE
       inner_var NUMBER;
    BEGIN
       inner_var := 'Mehmet'; -- Line A
       outer_var := outer_var || ' is';
    END;
    outer_var := outer_var || ' Zeynep';
EXCEPTION
    WHEN OTHERS THEN
       DBMS_OUTPUT.PUT_LINE(outer_var);
END;
```

Mark for Review
(1) Points

My

My name (*)

My name is

My name is Zeynep

Incorrect

Incorrect. Refer to Section 2.

24. Examine the following code. At Line A, we want to assign a value of 22 to the outer block's variable v_myvar. What code should we write at Line A?

```
<<outer_block>>
DECLARE
    v_myvar NUMBER;
BEGIN
    <<inner_block>>
    DECLARE
       v_myvar NUMBER := 15;
    BEGIN
       -- Line A
    END;
END;
```

Mark for Review
(1) Points

outer_block.v_myvar := 22; (*)

v_myvar := 22;

<<outer_block>>.v_myvar := 22;

v_myvar(outer_block) := 22;

We cannot reference the outer block's variable because both variables have the same name

Correct Correct

25. A collection is a composite data type. True or False? Mark for Review
(1) Points

True (*)

False

Incorrect Incorrect. Refer to Section 2.

26. What is the data type of the variable V_DEPT_TABLE in the following declaration?

```
DECLARE
TYPE dept_table_type IS TABLE OF departments%ROWTYPE INDEX BY PLS_INTEGER;
v_dept_table dept_table_type; ...
```

Mark for Review
(1) Points

Scalar

Composite (*)

LOB

Incorrect Incorrect. Refer to Section 2.

27. _____ are meant to store large amounts of data. Mark for Review
(1) Points

Variables

Scalar data types

LOBs (*)

Correct Correct

28. Variables can be assigned a value in both the Executable and Declaration sections of a PL/SQL program. True or False? Mark for Review (1) Points

True (*)

False

Correct Correct

29. Evaluate the following declaration. Determine whether or not it is legal.

```
DECLARE
maxsalary NUMBER(7) = 5000;
```

Mark for Review (1) Points

Correct.

Not correct. (*)

Correct Correct

30. Variables can be used in the following ways in a PL/SQL block. (Choose two.) Mark for Review (1) Points

(Choose all correct answers)

To store data values. (*)

To rename tables and columns.

To refer to a single data value several times. (*)

To comment code.

Incorrect Incorrect. Refer to Section 2.

31. When a variable is defined using the NOT NULL keywords, the variable must

contain a value. True or False? Mark for Review
(1) Points

True (*)

False

Correct Correct

32. When a variable is defined using the CONSTANT keyword, the value of the variable cannot change. True or False? Mark for Review
(1) Points

True (*)

False

Correct Correct

33. Single row character functions are valid SQL functions in PL/SQL. True or False? Mark for Review
(1) Points

True (*)

False

Correct Correct

34. Which of the following are disadvantages of implicit data type conversions? (Choose two.) Mark for Review
(1) Points

(Choose all correct answers)

The code is harder to read and understand (*)

You cannot store alphabetic characters in a variable of data type NUMBER

If Oracle changes the conversion rules in the future, your code may not work any more (*)

Oracle cannot implicitly convert a number value to a character string

Incorrect Incorrect. Refer to Section 2.

35. The DECODE function is available in PL/SQL procedural statements. True or False? Mark for Review
(1) Points

True

False (*)

Correct Correct

36. TO_NUMBER, TO_CHAR, and TO_DATE are all examples of: Mark for Review
(1) Points

Implicit conversion functions

Explicit conversion functions (*)

Character functions

Operators

Correct Correct

37. PL/SQL can convert a VARCHAR2 value containing alphabetic characters to a NUMBER value. True or False? Mark for Review
(1) Points

True

False (*)

Correct Correct

38. What is the output when the following program is executed?

```
set serveroutput on
DECLARE
    a VARCHAR2(10) := '333';
    b VARCHAR2(10) := '444';
    c PLS_INTEGER;
```

```
    d VARCHAR2(10);
BEGIN
    c := TO_NUMBER(a) + TO_NUMBER(b);
    d := a || b;
    DBMS_OUTPUT.PUT_LINE(c);
    DBMS_OUTPUT.PUT_LINE(d);
END;
```

Mark for Review
(1) Points

Nothing. The code will result in an error.

c=777 and d=333444 (*)

c=777 and d=777

c=333444 and d=777

Incorrect Incorrect. Refer to Section 2.

39. Examine the following code. What is the final value of V_MYBOOL ?

```
DECLARE
    v_mynumber NUMBER;
    v_mybool BOOLEAN ;
BEGIN
    v_mynumber := 6;
    v_mybool := (v_mynumber BETWEEN 10 AND 20);
    v_mybool := NOT (v_mybool);
END;
```

Mark for Review
(1) Points

True (*)

False

Incorrect Incorrect. Refer to Section 2.

40. What is wrong with this assignment statement?

```
myvar :=        'To be or not to be';
        'That is the question';
```

Mark for Review
(1) Points

An assignment statement must be a single line of code

Nothing is wrong, the statement is fine

An assignment statement must have a single semicolon at the end (*)

"myvar" is not a valid name for a variable

Character literals should not be enclosed in quotes

Incorrect Incorrect. Refer to Section 2.

41. Given this first section of code:

```
DECLARE
    v_result employees.salary%TYPE;
BEGIN
```

Which statement will always return exactly one value?

Mark for Review

(1) Points

```
SELECT salary
INTO v_result
FROM employees;
```

```
SELECT salary
INTO v_result
FROM employees
WHERE last_name ='Smith';
```

```
SELECT salary
INTO v_result
FROM employees
WHERE department_id = 80;
```

```
SELECT SUM(salary)
INTO v_result
FROM employees;
```

(*)

Incorrect Incorrect. Refer to Section 3.

42. Which rows will be deleted from the EMPLOYEES table when the following code is executed?

```
DECLARE
    salary employees.salary%TYPE := 12000;
BEGIN
    DELETE FROM employees
```

```
    WHERE salary > salary;
END;
```

Mark for Review
(1) Points

All rows whose SALARY column value is greater than 12000.

All rows in the table.

No rows. (*)

All rows whose SALARY column value is equal to 12000.

Incorrect Incorrect. Refer to Section 3.

43. The following code will return the last name of the employee whose employee id is equal to 100: True or False?

```
DECLARE
    v_last_name employees.last_name%TYPE;
    employee_id employees.employee_id%TYPE := 100;
BEGIN
    SELECT last_name INTO v_last_name
    FROM employees
    WHERE employee_id = employee_id;
END;
```

Mark for Review
(1) Points

True

False (*)

Correct Correct

44. A variable is declared as:

```
DECLARE
    v_holdit employees.last_name%TYPE;
BEGIN ...
```

Which of the following is a correct use of the INTO clause?
Mark for Review
(1) Points

```
SELECT *
INTO v_holdit
FROM employees;
```

```
        SELECT last_name
INTO v_holdit
FROM employees;
```

```
        SELECT last_name
INTO v_holdit
FROM employees
WHERE employee_id=100;
```

(*)

```
        SELECT salary
INTO v_holdit
FROM employees
WHERE employee_id=100;
```

Incorrect Incorrect. Refer to Section 3.

45. Which of the following is NOT a good guideline for retrieving data in PL/SQL? Mark for Review
(1) Points

Declare the receiving variables using %TYPE

The WHERE clause is optional in nearly all cases. (*)

Specify the same number of variables in the INTO clause as database columns in the SELECT clause.

THE SELECT statement should fetch exactly one row.

Correct Correct

46. How many DML statements can be included in a single transaction? Mark for Review
(1) Points

Only one

None. A transaction cannot include DML statements.

A maximum of four DML statements

As many as needed (*)

Incorrect

Incorrect. Refer to Section 3.

47. The following anonymous block of code is run:

```
BEGIN
    INSERT INTO countries (id, name)
    VALUES ('XA', 'Xanadu');
    INSERT INTO countries (id, name)
    VALUES ('NV','Neverland');
    COMMIT;
    COMMIT;
    ROLLBACK;
END;
```

What happens when the block of code finishes?

Mark for Review

(1) Points

You have nothing new; the last ROLLBACK undid the INSERTs.

You have the rows added twice; there are four new rows.

You have the two new rows added. (*)

You get an error; you cannot COMMIT twice in a row.

Incorrect

Incorrect. Refer to Section 3.

48. A PL/SQL block includes the following statement:

```
SELECT last_name INTO v_last_name
FROM employees
WHERE employee_id=100;
```

What is the value of SQL%ISOPEN immediately after the SELECT statement is executed?

Mark for Review

(1) Points

True

False (*)

Null

Error. That attribute does not apply for implicit cursors.

Incorrect Incorrect. Refer to Section 3.

49. Assume there are 5 employees in Department 10. What happens when the following statement is executed?

```
UPDATE employees
SET salary=salary*1.1;
```

Mark for Review
(1) Points

All employees get a 10% salary increase. (*)

No rows are modified because you did not specify "WHERE department_id=10"

A TOO_MANY_ROWS exception is raised.

An error message is displayed because you must use the INTO clause to hold the new salary.

Incorrect Incorrect. Refer to Section 3.

50. Which SQL statement can NOT use an implicit cursor? Mark for Review
(1) Points

A DELETE statement

An UPDATE statement

A SELECT statement that returns multiple rows (*)

A SELECT statement that returns one row

Correct Correct
1. Examine the following code:

```
DECLARE
    v_salary NUMBER(6);
    v_constant NUMBER(6) := 15000;
    v_result VARCHAR(6); := 'MIDDLE';
BEGIN
    IF v_salary != v_constant THEN
        v_result := 'HIGH';
    ELSE
        v_result := 'LOW';
    END IF;
END;
```

What is the final value of v_result?
Mark for Review
(1) Points

HIGH

LOW (*)

MIDDLE

Null

Correct Correct

2. Examine the following code:

```
DECLARE
    a VARCHAR2(6) := NULL;
    b VARCHAR2(6) := NULL;
BEGIN
    IF a = b THEN
       DBMS_OUTPUT.PUT_LINE('EQUAL');
    ELSIF a != b THEN
       DBMS_OUTPUT.PUT_LINE('UNEQUAL');
    ELSE
       DBMS_OUTPUT.PUT_LINE('OTHER');
    END IF;
END;
```

Which word will be displayed?
Mark for Review
(1) Points

UNEQUAL

EQUAL

Nothing will be displayed

OTHER (*)

Incorrect Incorrect. Refer to Section 4.

3. How many ELSIF statements are you allowed to have in a compound IF statement? Mark for Review
(1) Points

Only one

As many as you want (*)

They must match the same number as the number of ELSE statements.

None; the command is ELSE IF;

Incorrect Incorrect. Refer to Section 4.

4. What is the correct form of a simple IF statement? Mark for Review
(1) Points

IF condition THEN statement;

IF condition THEN statement;
END IF; (*)

IF condition;
THEN statement;
END IF;

IF condition
THEN statement
ENDIF;

Correct Correct

5. You need to execute a set of statements 10 times, increasing a counter by 1 each time. Which of the following PL/SQL constructs can do this? (Choose three) Mark for Review
(1) Points

(Choose all correct answers)

IF ... THEN ... ELSE

A WHILE loop (*)

CASE ... WHEN ... THEN

A FOR loop (*)

A basic loop (*)

Incorrect Incorrect. Refer to Section 4.

6. What kind of statement is best suited for displaying the multiplication table for "sixes": 6x1=6, 6x2=12 ... 6x12=72? Mark for Review (1) Points

CASE expression

IF statement

CASE statement

LOOP statement (*)

Incorrect Incorrect. Refer to Section 4.

7. Which kind of loop is this?

```
v_count := 1;
LOOP
    v_count := v_count + 1;
    EXIT WHEN i > 20;
END LOOP;
```

Mark for Review (1) Points

FOR loop

IF-THEN loop

Basic loop (*)

WHILE loop

CASE loop

Correct Correct

8. Which one of these tasks is best done using a LOOP statement? Mark for Review (1) Points

Assigning a letter grade to a numerical score

Calculating and displaying the sum of all integers from 1 to 100 (*)

Testing if a condition is true, false or null

Fetching and displaying an employee's last name from the database

Incorrect Incorrect. Refer to Section 4.

9. A PL/SQL block contains the following code:

```
v_counter := 1;
LOOP
    EXIT WHEN v_counter=5;
END LOOP;
v_counter := v_counter + 1;
```

What is the value of V_COUNTER after the loop is finished?
Mark for Review
(1) Points

5

6

1

This is an infinite loop; the loop will never finish. (*)

Correct Correct

10. Which one of these is NOT a kind of loop? Mark for Review
(1) Points

ASCENDING loop (*)

FOR loop

Basic loop

WHILE loop

Incorrect Incorrect. Refer to Section 4.

11. What will be the value of v_sal_desc after the following code is executed?

```
DECLARE
    v_salary NUMBER(6,2) := NULL;
    v_sal_desc VARCHAR2(10);
BEGIN
    CASE
       WHEN v_salary < 10000 THEN v_sal_desc := 'Low Paid';
       WHEN v_salary >= 10000 THEN v_sal_desc := 'High Paid';
    END CASE;
END;
```

Mark for Review
(1) Points

High Paid

Low Paid

Null

The code will fail and return an exception (*)

Incorrect Incorrect. Refer to Section 4.

12. You want to assign a value to v_result which depends on the value of v_grade: if v_grade = 'A' set v_result to 'Very Good' and so on.

```
DECLARE
    v_grade CHAR(1);
    v_result VARCHAR2(10);
BEGIN
    v_result :=
       CASE v_grade
```

The next line should be
Mark for Review
(1) Points

WHEN v_grade = 'A' THEN 'Very Good'

WHEN 'A' THEN 'Very Good';

WHEN 'A' THEN v_result := 'Very Good';

WHEN 'A' THEN 'Very Good' (*)

Incorrect Incorrect. Refer to Section 4.

13. What will be the value of variable c after the following code is executed?

```
DECLARE
    a BOOLEAN := TRUE;
    b BOOLEAN := FALSE;
    c NUMBER;
BEGIN
    c :=
        CASE
            WHEN a AND b THEN 10
            WHEN NOT a THEN 20
            WHEN a OR b THEN 30
            ELSE 40
        END;
END;
```

Mark for Review
(1) Points

30 (*)

20

40

10

Incorrect

Incorrect. Refer to Section 4.

14. What will be the value of variable c after the following code is executed?

```
DECLARE
    a BOOLEAN := TRUE;
    b BOOLEAN := NULL;
    c NUMBER;
BEGIN
    IF a AND b THEN c := 2;
       ELSIF a OR b THEN c := 0;
       ELSE c := 1;
    END IF;
END;
```

Mark for Review
(1) Points

1

Null

0 (*)

2

Incorrect Incorrect. Refer to Section 4.

15. What value will v_answer contain after the following code is executed?

```
DECLARE
    v_age NUMBER:= 18;
    v_answer VARCHAR2(10);
BEGIN
    v_answer :=
       CASE
          WHEN v_age < 25 THEN 'Young'
          WHEN v_age = 18 THEN 'Exactly 18'
          ELSE 'Older'
       END CASE;
END;
```

Mark for Review
(1) Points

Exactly 18

Young (*)

Null

Older

Correct Correct

16. Examine the following code:

```
DECLARE
v_bool BOOLEAN := FALSE;
v_counter NUMBER(4) := 0;
BEGIN
... Line A
?
END;
```

Which of the following is NOT valid at line A?
Mark for Review
(1) Points

WHILE NOT v_boolean LOOP

WHILE v_boolean AND v_counter < 6 LOOP

WHILE v_counter > 8 LOOP

WHILE v_counter IN 1..5 LOOP (*)

Incorrect Incorrect. Refer to Section 4.

17. In a FOR loop, an implicitly declared counter automatically increases or decreases with each iteration. True or False? Mark for Review (1) Points

True (*)

False

Incorrect Incorrect. Refer to Section 4.

18. Which statement best describes when a FOR loop should be used? Mark for Review (1) Points

When the number of iterations is known (*)

When testing the value in a Boolean variable

When the controlling condition must be evaluated at the start of each iteration

Incorrect Incorrect. Refer to Section 4.

19. Which statement best describes when a WHILE loop shouild be used? Mark for Review (1) Points

When the number of iterations is known

When repeating a sequence of statements until the controlling condition is no longer true (*)

When assigning a value to a Boolean variable

When testing whether a variable is null

Incorrect Incorrect. Refer to Section 4.

20. In a WHILE loop, the statements inside the loop must execute at least once. True or False? Mark for Review
(1) Points

True

False (*)

Incorrect Incorrect. Refer to Section 4.

21. What will happen when the following code is executed?

```
BEGIN
FOR i in 1 ..3 LOOP
    DBMS_OUTPUT.PUT_LINE (i);
    i := i + 1;
END LOOP;
END;
```

Mark for Review
(1) Points

It will display 1, 2, 3.

It will display 2, 3, 4.

It will result in an error because you cannot modify the counter in a FOR loop. (*)

It will result in an error because the counter was not explicitly declared.

Incorrect Incorrect. Refer to Section 4.

22. What kinds of loops can be nested? Mark for Review
(1) Points

BASIC loops

WHILE loops

FOR loops

All of the above (*)

Incorrect Incorrect. Refer to Section 4.

23. When coding two nested loops, both loops must be of the same

type. For example, you cannot code a FOR loop inside a WHILE loop. True or False?
Mark for Review
(1) Points

True

False (*)

Correct Correct

24. In the following code fragment, you want to exit from the outer loop at Line A if v_number = 6. Which statement would you write on Line A?

```
<<big_loop>>
WHILE condition_1 LOOP
    <<small_loop>>
    FOR i IN 1..10 LOOP
       DBMS_OUTPUT.PUT_LINE(i);
       -- Line A
    END LOOP;
END LOOP;
```

Mark for Review
(1) Points

IF v_number = 6 THEN EXIT;

EXIT outer_loop WHEN v_number = 6;

EXIT big_loop WHEN v_number = 6; (*)

EXIT small_loop WHEN v_number = 6;

Incorrect Incorrect. Refer to Section 4.

25. Examine the following code:

```
BEGIN
FOR i IN 1..5 LOOP
FOR j IN 1..8 LOOP
EXIT WHEN j = 7;
DBMS_OUTPUT.PUT_LINE(i || j);
END LOOP;
END LOOP;
END;
```

How many lines of output will be displayed when this code is executed? Mark for Review
(1) Points

35

6

30 (*)

40

Correct Correct

Section 5

26. What is wrong with the following code?

```
DECLARE
CURSOR emp_curs(p_dept_id NUMBER) IS
SELECT * FROM employees WHERE department_id = p_dept_id;
BEGIN
FOR dept_rec IN (SELECT * FROM departments) LOOP
DBMS_OUTPUT.PUT_LINE(dept_rec.department_name);
FOR emp_rec IN emp_curs(dept_rec.department_id) LOOP
DBMS_OUTPUT.PUT_LINE(emp_rec.last_name);
END LOOP;
END LOOP;
END;
```

Mark for Review
(1) Points

The DEPARTMENTS cursor must be declared with a parameter.

You cannot use a cursor with a subquery in nested loops.

You cannot use two different kinds of loop in a single PL/SQL block.

EMP_CURS should not be DECLAREd explicitly; it should be coded as a subquery in a cursor FOR loop.

Nothing is wrong. The block will execute successfully and display all departments and the employees in those departments. (*)

Correct Correct

27. When using multiple nested cursors, what kinds of loops can you use? Mark for Review
(1) Points

Cursor FOR loops only.

Basic loops only.

WHILE loops only.

None of the above.

All of the above. (*)

Incorrect Incorrect. Refer to Section 5.

28. You want to display all locations, and the departments in each location. Examine the following code:

```
DECLARE
CURSOR loc_curs IS SELECT * FROM locations;
CURSOR dept_curs(p_loc_id NUMBER) IS
SELECT * FROM departments WHERE location_id = p_loc_id;
BEGIN
FOR loc_rec IN loc_curs LOOP
DBMS_OUTPUT.PUT_LINE(loc_rec.city);
FOR dept_rec IN dept_curs(-- Point A --) LOOP
DBMS_OUTPUT.PUT_LINE(dept_rec.department_name);
END LOOP;
END LOOP;
END;
```

What should you code at Point A? Mark for Review
(1) Points

p_loc_id

location_id

null

LOOP ... END LOOP;

loc_rec.location_id (*)

Incorrect Incorrect. Refer to Section 5.

29. Assume that you have declared a cursor called C_EMP. Which of the following statements about C_EMP is correct? (Choose two.) Mark for Review
(1) Points

(Choose all correct answers)

You can use c_emp%NOTFOUND to exit a loop. (*)

You can fetch rows when c_emp%ISOPEN evaluates to FALSE.

You can use c_emp%ROWCOUNT to return the number of rows returned by the cursor so far. (*)

You can use c_emp%FOUND after the cursor is closed.

Correct Correct

30. Which of the following statements about the %ISOPEN cursor attribute is true? Mark for Review
(1) Points

You can issue the %ISOPEN cursor attribute only when a cursor is open.

You can issue the %ISOPEN cursor attribute only when more than one record is returned.

You can issue the %ISOPEN cursor attribute when a cursor is open or closed. (*)

If a cursor is open, then the value of %ISOPEN is false.

Incorrect Incorrect. Refer to Section 5.

31. The DEPARTMENTS table contains four columns. Examine the following code:

```
DECLARE
    CURSOR dept_curs IS
       SELECT * FROM departments;
    v_dept_rec dept_curs%ROWTYPE;
BEGIN
    OPEN dept_curs;
    FETCH dept_curs INTO v_dept_rec;
    ...
```

Which one of the following statements is true?
Mark for Review
(1) Points

v_dept_rec contains the first four rows of the departments table.

The FETCH will fail because the structure of v_dept_rec does not match the structure of the cursor.

v_dept_rec contains the first row of the departments table. (*)

The block will fail because the declaration of v_dept_rec is invalid.

Correct Correct

32. Which of the following cursor attributes is set to the total number of rows returned so far? Mark for Review
(1) Points

%ISOPEN

%NOTFOUND

%FOUND

%ROWCOUNT (*)

Incorrect Incorrect. Refer to Section 5.

33. Examine the following code fragment:

```
DECLARE
    CURSOR emp_curs IS
       SELECT first_name, last_name FROM employees;
    v_emp_rec emp_curs%ROWTYPE;
BEGIN
    ...
    FETCH emp_curs INTO v_emp_rec;
    DBMS_OUTPUT.PUT_LINE(... Point A ...);
   ...
```

To display the fetched last name, what should you code at Point A?
Mark for Review
(1) Points

v_emp_rec.last_name (*)

v_emp_rec(last_name)

v_emp_rec

last_name

None of the above

Incorrect Incorrect. Refer to Section 5.

34. Which of the following cursor attributes evaluates to TRUE if the cursor is open? Mark for Review

(1) Points

%ISOPEN (*)

%NOTFOUND

%FOUND

%ROWCOUNT

Incorrect Incorrect. Refer to Section 5.

35. The employees table contains 20 rows. What will happen when the following code is executed?

```
DECLARE
  &nbspCURSOR emp_curs IS
     &nbspSELECT job_id FROM employees;
    v_job_id employees.job_id%TYPE;
BEGIN
    OPEN emp_curs;
    LOOP
       FETCH emp_curs INTO v_job_id;
       DBMS_OUTPUT.PUT_LINE(v_job_id);
       EXIT WHEN emp_curs%NOTFOUND;
    END LOOP;
    CLOSE emp_curs;
END;
```

Mark for Review
(1) Points

20 job_ids will be displayed.

The block will fail and an error message will be displayed.

21 rows of output will be displayed; the first job_id will be displayed twice.

21 rows of output will be displayed; the last job_id will be displayed twice. (*)

Correct Correct

36. An implicit cursor can be used for a multiple-row SELECT statement. True or False? Mark for Review
(1) Points

True

False (*)

Correct Correct

37. Place the following statements in the correct sequence:

1. OPEN my_curs;
2. CLOSE my_curs;
3. CURSOR my_curs IS SELECT my_column FROM my_table;
4. FETCH my_curs INTO my_variable;

Mark for Review
(1) Points

C,D,A,B

C,A,D,B (*)

A,C,D,B

C,A,B,D

Correct Correct

38. What will happen when the following code is executed?

```
DECLARE CURSOR emp_curs IS
    SELECT salary FROM employees;
    v_salary employees.salary%TYPE;
BEGIN
    OPEN emp_curs;
    FETCH emp_curs INTO v_salary;
    CLOSE emp_curs;
    FETCH emp_curs INTO v_salary;
END;
```

Mark for Review
(1) Points

The block will fail and an INVALID_CURSOR exception will be raised. (*)

The first employee row will be fetched twice.

The first two employee rows will be fetched.

The block will fail and a TOO_MANY_ROWS exception will be raised.

Incorrect Incorrect. Refer to Section 5.

39. After a cursor has been closed, it can be opened again in the same PL/SQL block. True or False? Mark for Review
(1) Points

True (*)

False

Incorrect Incorrect. Refer to Section 5.

40. For which type of SQL statement must you use an explicit cursor? Mark for Review
(1) Points

DML statements that process more than one row.

Queries that return more than one row. (*)

Data Definition Language (DDL) statements.

Queries that return a single row.

Incorrect Incorrect. Refer to Section 5.
41. What will happen when the following code is executed?

```
DECLARE
    CURSOR emp_curs IS
       SELECT salary FROM employees;
    v_salary employees.salary%TYPE;
BEGIN
    FETCH emp_curs INTO v_salary;
    DBMS_OUTPUT.PUT_LINE(v_salary);
    CLOSE emp_curs;
END;
```

Mark for Review
(1) Points

The first employee's salary will be fetched and displayed.

All employees' salaries will be fetched and displayed.

The execution will fail and an error message will be displayed. (*)

The lowest salary value will be fetched and displayed.

Correct Correct

42. Examine the following code:

```
DECLARE
    CURSOR emp_curs IS
       SELECT last_name, salary
       FROM employees
       ORDER BY salary;
    v_last_name employees.last_name%TYPE;
    v_salary employees.salary%TYPE;
BEGIN
...
```

Which of the following statements successfully opens the cursor and fetches the first row of the active set?
Mark for Review
(1) Points

```
        OPEN emp_curs;
FETCH emp_curs INTO v_last_name, v_salary;
```

(*)

```
        OPEN emp_curs;
FETCH emp_curs INTO v_salary, v_last_name;
```

```
        OPEN emp_curs;
FETCH FIRST emp_curs INTO v_last_name, v_salary;
```

```
        OPEN emp_curs;
FETCH emp_curs;
```

Incorrect Incorrect. Refer to Section 5.

43. The following code fragment shows a cursor FOR loop:

```
FOR emp_record IN emp_cursor LOOP ......
```

Which of the following do NOT need to be coded explicitly? (Choose three.)
Mark for Review
(1) Points

(Choose all correct answers)

OPEN emp_cursor; (*)

DECLARE CURSOR emp_cursor IS ...

emp_record emp_cursor%ROWTYPE; (*)

FETCH emp_cursor INTO emp_record; (*)

END LOOP;

Incorrect Incorrect. Refer to Section 5

44. What is wrong with the following code?

```
DECLARE
    CURSOR dept_curs IS SELECT * FROM departments;
BEGIN
    FOR dept_rec IN dept_curs LOOP
       DBMS_OUTPUT.PUT_LINE(dept_curs%ROWCOUNT || dept_rec.department_name);
    END LOOP;
    DBMS_OUTPUT.PUT_LINE(dept_rec.department_id);
END;
```

Mark for Review
(1) Points

The cursor DEPT_CURS has not been opened.

The implicitly declared record DEPT_REC cannot be referenced outside the cursor FOR loop. (*)

You cannot use %ROWCOUNT with a cursor FOR loop.

The cursor DEPT_CURS has not been closed.

Nothing is wrong, this code will execute successfully.

Incorrect Incorrect. Refer to Section 5

45. What is wrong with the following code?

```
BEGIN
    FOR emp_rec IN
       (SELECT * FROM employees WHERE ROWNUM < 10
          FOR UPDATE NOWAIT) LOOP
        DBMS_OUTPUT.PUT_LINE(emp_rec%ROWCOUNT || emp_rec.last_name):
    END LOOP;
END;
```

Mark for Review
(1) Points

You cannot use FOR UPDATE NOWAIT with a cursor FOR loop using a subquery.

You cannot reference %ROWCOUNT with a cursor FOR loop using a subquery. (*)

The field EMP_REC.LAST_NAME does not exist.

You cannot use ROWNUM with a cursor FOR loop.

The cursor has not been opened.

Correct Correct

46. User MARY has locked a row of the EMPLOYEES table. Now, user SAEED tries to open the following cursor:
CURSOR c IS
SELECT * FROM employees
FOR UPDATE WAIT 5;
What will happen when SAEED's session tries to fetch the row that MARY has locked?
Mark for Review
(1) Points

SAEED's session successfully fetches the first 5 rows and then waits indefinitely to fetch the 6th row.

SAEED's session waits for 5 seconds, and then raises an exception if MARY has not unlocked the row. (*)

SAEED's session waits for 5 seconds, then SAEED is disconnected from the database.

SAEED's session waits for 5 seconds, then MARY's session is rolled back.

SAEED's session waits for 5 minutes, and then raises an exception if MARY has not unlocked the row.

Incorrect Incorrect. Refer to Section 5.

47. User TOM has locked a row in the WORKERS table. Now, user DICK wants to open the following cursor:
CURSOR c IS
SELECT * FROM workers FOR UPDATE NOWAIT;
What will happen when DICK opens the cursor and tries to fetch rows? Mark for Review
(1) Points

TOM's session is rolled back. DICK's session successfully fetches rows from the cursor.

DICK's session waits indefinitely.

Both sessions wait for a few seconds; then the system breaks all locks and both sessions raise an exception.

DICK's session immediately raises an exception. (*)

The c%NOWAIT attribute is set to TRUE.

Incorrect Incorrect. Refer to Section 5.

48. You want to declare a cursor which locks each row fetched by the cursor. Examine the following code:
DECLARE
CURSOR emp_curs IS
SELECT * FROM employees
FOR -- Point A
Which of the following can NOT be coded at Point A? Mark for Review
(1) Points

UPDATE;

UPDATE OF salary;

UPDATE OF employees; (*)

UPDATE NOWAIT;

Incorrect Incorrect. Refer to Section 5.

49. What is one of the advantages of using parameters with a cursor? Mark for Review
(1) Points

You can use a cursor FOR loop.

You can declare the cursor FOR UPDATE.

You do not need to DECLARE the cursor at all.

You can use a single cursor to fetch a different set of rows each time the cursor is opened. (*)

It will execute much faster than a cursor without parameters.

Correct Correct

50. There are 12 distinct JOB_IDs in the EMPLOYEES table. You need to write some PL.SQL code to fetch and display all the employees with a specific JOB_ID. The chosen JOB_ID can be different each time the code is executed. What is the best way to do this? Mark for Review
(1) Points

Write 12 separate PL/SQL blocks, each declaring a cursor with a different JOB_ID in the WHERE clause.

Write a single PL/SQL block which declares 12 cursors, one for each distinct value of JOB_ID.

Write a single PL/SQL block which declares one cursor using a parameter for the JOB_ID. (*)

Write a single PL/SQL block which uses a cursor to fetch all the employee rows, with an IF statement to decide which of the fetched rows to display.

Incorrect Incorrect. Refer to Section 5.

1. A program which specifies a list of operations to be performed sequentially to achieve the desired result can be called: Mark for Review
(1) Points

declarative

nondeclarative

procedural (*)

low level

Correct

2. SQL is a common access language for many types of databases, including Oracle. True or False? Mark for Review
(1) Points

True (*)

False

Correct

3. Which of the following statements about PL/SQL and SQL is true? Mark for Review
(1) Points

PL/SQL and SQL are both ANSI-compliant.

PL/SQL and SQL can be used with many types of databases, including Oracle.

PL/SQL and SQL are both Oracle proprietary programming languages.

PL/SQL allows basic program logic and control flow to be combined with SQL statements. (*)

Correct

4. Which of the following can you use PL/SQL to do? Mark for Review
(1) Points

Update data (DML)

Develop Web applications using the Web Application Toolkit

Manage database security

Create customized reports

All of the above (*)

Correct

5. PL/SQL can be used not only with an Oracle database, but also with any kind of relational database. True or False? Mark for Review
(1) Points

True

False (*)

Correct

6. Which of the following statements about exception handling in PL/SQL is false? Mark for Review
(1) Points

You can prepare for database exceptions by creating exception handlers.

You can prepare for application exceptions by creating exception handlers.

Exception handling code tells your program what to do when an error is encountered.

Exception handling code can be grouped together in a PL/SQL block.

None of the above (*)

Incorrect. Refer to Section 1.

7. What kind of block is defined by the following PL/SQL code?

```
BEGIN
    DBMS_OUTPUT.PUT_LINE('My first quiz');
END;
```

Mark for Review
(1) Points

procedure

subroutine

function

anonymous (*)

Incorrect. Refer to Section 1.

8. Which keywords must be included in every PL/SQL block? (Choose two.) Mark for Review
(1) Points

(Choose all correct answers)

DECLARE

END; (*)

EXCEPTION

BEGIN (*)

DBMS_OUTPUT.PUT_LINE

Incorrect. Refer to Section 1.

9. Given below are the parts of a PL/SQL block:

1. END;
2. EXCEPTION
3. DECLARE
4. BEGIN

Arrange the parts in order.

Mark for Review
(1) Points

2,1,4,3

3,4,2,1 (*)

3,2,4,1

4,3,2,1

Correct

10. What is the purpose of using DBMS_OUTPUT.PUT_LINE in a PL/SQL block? Mark for Review
(1) Points

To perform conditional tests

To allow a set of statements to be executed repeatedly

To display results to check if our code is working correctly (*)

To store new rows in the database

Correct

11. Errors are handled in the Exception part of the PL/SQL block. True or False? Mark for Review
(1) Points

True (*)

False

Correct

12. In which part of the PL/SQL block are declarations of variables defined? Mark for Review
(1) Points

Executable

Exception

Declarative (*)

Definition

Correct

13. Which statements are optional in a PL/SQL block? (Choose two.) Mark for Review
(1) Points

(Choose all correct answers)

DECLARE (*)

BEGIN

EXCEPTION (*)

END;

Correct

Section 2

14. When you use a function to convert data types in a PL/SQL program, it is called _____ conversion. Mark for Review
(1) Points

Explicit (*)

Implicit

TO_CHAR

Correct

15. What is the output when the following program is executed?

```
set serveroutput on
DECLARE
    a VARCHAR2(10) := '333';
    b VARCHAR2(10) := '444';
    c PLS_INTEGER;
    d VARCHAR2(10);
BEGIN
    c := TO_NUMBER(a) + TO_NUMBER(b);
    d := a || b;
    DBMS_OUTPUT.PUT_LINE(c);
    DBMS_OUTPUT.PUT_LINE(d);
END;
```

Mark for Review
(1) Points

Nothing. The code will result in an error.

c=777 and d=333444 (*)

c=777 and d=777

c=333444 and d=777

Correct

16. Which of the following are disadvantages of implicit data type conversions? (Choose two.) Mark for Review
(1) Points

(Choose all correct answers)

The code is harder to read and understand (*)

You cannot store alphabetic characters in a variable of data type NUMBER

If Oracle changes the conversion rules in the future, your code may not work any more (*)

Oracle cannot implicitly convert a number value to a character string

Correct

17. Examine the following code:

```
1 DECLARE
2 x NUMBER;
3 BEGIN
4 x:= '300';
5 END;
```

After line 4, what is the value of x?

Mark for Review
(1) Points

'300'

300 (*)

NULL

Correct

18. Single row character functions are valid SQL functions in PL/SQL. True or False? Mark for Review
(1) Points

True (*)

False

Correct

19. The implicit data type conversion at Point A may not work correctly. Why not?

```
DECLARE
    v_mydate DATE;
BEGIN
    V_MYDATE := '29-Feb-04'; -- Point A
END;
```

Mark for Review

(1) Points

There are only 28 days in February

Oracle cannot implicitly convert a character string to a date, even if the string contains a valid date value

If the database language is not English, 'Feb' has no meaning. (*)

V_MYDATE has been entered in uppercase

Correct

20. What is wrong with this assignment statement?
myvar := 'To be or not to be';
'That is the question';
Mark for Review
(1) Points

An assignment statement must be a single line of code

Nothing is wrong, the statement is fine

An assignment statement must have a single semicolon at the end (*)

"myvar" is not a valid name for a variable

Character literals should not be enclosed in quotes

Correct

21. The DECODE function is available in PL/SQL procedural statements. True or False? Mark for Review
(1) Points

True

False (*)

Correct

22. You need to declare a variable to hold a value which has been read from the SALARY column of the EMPLOYEES table. Which of the following is an advantage of declaring the variable as: employees.salary%TYPE ? Mark for

Review
(1) Points

It is shorter than coding NUMBER(8,2)

If the SALARY column is ALTERed later, the PL/SQL code need not be changed. (*)

It executes much faster than using NUMBER(8,2)

It allows the software to perform implicit data type conversions.

Correct

23. Which of the following declarations is invalid? Mark for Review
(1) Points

v_count PLS_INTEGER:=0;

college_name VARCHAR2(20):='Harvard';

v_pages CONSTANT NUMBER; (*)

v_start_date DATE := sysdate+1;

Correct

24. If you are using the %TYPE attribute, you can avoid hard coding the: Mark for Review
(1) Points

Data type (*)

Table name

Column name

Constraint

Incorrect. Refer to Section 2.

PlSQL feedback midterm semister 1 part2

25. Is the following variable declaration correct or not ?

```
DECLARE
display_qty CONSTANT NUMBER;
```

Mark for Review
(1) Points

Correct.

Not correct. (*)

Incorrect. Refer to Section 2.

26. Variables can be assigned a value in both the Executable and Declaration sections of a PL/SQL program. True or False? Mark for Review
(1) Points

True (*)

False

Correct

27. When a variable is defined using the CONSTANT keyword, the value of the variable cannot change. True or False? Mark for Review
(1) Points

True (*)

False

Correct

28. Identify which of the following assignment statements are valid. (Choose three.) Mark for Review
(1) Points

(Choose all correct answers)

v_last_name := Chandra;

v_blackout_date := '31-DEC-2006'; (*)

v_population := 333444; (*)

v_music_type := 'ROCK'; (*)

Correct

29. Assignment statements can continue over several lines in PL/SQL. True or False? Mark for Review
(1) Points

True (*)

False

Correct

30. Valid identifiers begin with a Mark for Review
(1) Points

Number

Letter (*)

Special character

Correct

31. Which of the following are valid identifiers? (Choose two.) Mark for Review
(1) Points

(Choose all correct answers)

yesterday (*)

yesterday's date

number_of_students_in_the_class

v$testresult (*)

#students

Incorrect. Refer to Section 2.

32. Which of the following are PL/SQL lexical units? (Choose two.) Mark for Review
(1) Points

(Choose all correct answers)

Identifiers (*)

Table Columns

Reserved Words (*)

Anonymous Blocks

SQL Workshop

Correct

33. What is the data type of the variable V_DEPT_TABLE in the following declaration?

```
DECLARE
TYPE dept_table_type IS TABLE OF departments%ROWTYPE INDEX BY PLS_INTEGER;
v_dept_table dept_table_type; ...
```

Mark for Review
(1) Points

Scalar

Composite (*)

LOB

Correct

34. _____ are meant to store large amounts of data. Mark for Review
(1) Points

Variables

Scalar data types

LOBs (*)

Correct

35. A collection is a composite data type. True or False? Mark for Review
(1) Points

True (*)

False

Correct

36. When nested blocks are used, which blocks can or must be labeled? Mark for Review
(1) Points

The inner block must be labeled, the outer block can be labeled.

Both blocks must be labeled

Nested blocks cannot be labeled

The outer block must be labeled if it is to be referred to in the inner block. (*)

Correct

37. When an exception occurs within a PL/SQL block, the remaining statements in the executable section of the block are skipped. True or False? Mark for Review
(1) Points

True (*)

False

Correct

38. What will be displayed when the following code is executed?

```
DECLARE
    x VARCHAR2(6) := 'Chang';
BEGIN
```

```
    DECLARE
       x VARCHAR2(12) := 'Susan';
    BEGIN
       x := x || x;
    END;
    DBMS_OUTPUT.PUT_LINE(x);
END;
```

Mark for Review
(1) Points

Susan

Chang (*)

ChangChang

SusanChang

The code will fail with an error

Incorrect. Refer to Section 2.

39. An exception occurs within the inner block of two nested blocks. The inner block does not have an EXCEPTION section. What always happens?
Mark for Review
(1) Points

Both blocks fail and an error message is displayed by the calling environment

The exception is propagated to the outer block (*)

Oracle automatically tries to re-execute the inner block

The user's database session is automatically disconnected

Incorrect. Refer to Section 2.

40. Examine the following code. Line A causes an exception. What will be displayed when the block is executed?

```
DECLARE
    var_a NUMBER := 6;
    var_b DATE;
BEGIN
    var_a := var_a * 2;
    var_b := '28 December 2006'; -- Line A
    var_a := var_a * 2;
EXCEPTION
```

```
    WHEN OTHERS THEN
       DBMS_OUTPUT.PUT_LINE(var_a);
END;
```

Mark for Review
(1) Points

12 (*)

24

6

Nothing will be displayed

Correct

41. The following anonymous block of code is run:

```
BEGIN
    INSERT INTO countries (id, name)
    VALUES ('XA', 'Xanadu');
    SAVEPOINT XA;
    INSERT INTO countries (id, name)
    VALUES ('NV','Neverland');
    COMMIT;
    ROLLBACK TO XA;
END;
```

What happens when the block of code finishes?

Mark for Review
(1) Points

No data is inserted and no errors occur.

No data is inserted and an error occurs

Two rows are inserted and no errors occur.

Two rows are inserted and an error occurs. (*)

Correct

42. The following anonymous block of code is run:

```
BEGIN
    INSERT INTO countries (id, name)
    VALUES ('XA', 'Xanadu');
    INSERT INTO countries (id, name)
```

```
    VALUES ('NV','Neverland');
    COMMIT;
    COMMIT;
    ROLLBACK;
END;
```

What happens when the block of code finishes?

Mark for Review
(1) Points

You have nothing new; the last ROLLBACK undid the INSERTs.

You have the rows added twice; there are four new rows.

You have the two new rows added. (*)

You get an error; you cannot COMMIT twice in a row.

Correct

43. Which of the following is NOT a good guideline for retrieving data in PL/SQL? Mark for Review
(1) Points

Declare the receiving variables using %TYPE

The WHERE clause is optional in nearly all cases. (*)

Specify the same number of variables in the INTO clause as database columns in the SELECT clause.

THE SELECT statement should fetch exactly one row.

Correct

44. The following code will return the last name of the employee whose employee id is equal to 100: True or False?

```
DECLARE
    v_last_name employees.last_name%TYPE;
    employee_id employees.employee_id%TYPE := 100;
BEGIN
    SELECT last_name INTO v_last_name
    FROM employees
    WHERE employee_id = employee_id;
END;
```

Mark for Review
(1) Points

True

False (*)

Correct

45. A variable is declared as:

```
DECLARE
    v_holdit employees.last_name%TYPE;
BEGIN ...
```

Which of the following is a correct use of the INTO clause?

Mark for Review
(1) Points

```
SELECT *
INTO v_holdit
FROM employees;
```

```
SELECT last_name
INTO v_holdit
FROM employees;
```

```
SELECT last_name
INTO v_holdit
FROM employees
WHERE employee_id=100;
```
(*)

```
SELECT salary
INTO v_holdit
FROM employees
WHERE employee_id=100;
```

Correct

46. Which one of these SQL statements can be directly included in a PL/SQL executable block? Mark for Review
(1) Points

```
DELETE FROM employees
WHERE department_id=60;
```
(*)

SELECT salary FROM employees

```
WHERE department_id=60;
```

```
CREATE TABLE new_emps (last_name VARCHAR2(10), first_name VARCHAR2(10));
```

```
DROP TABLE locations;
```

Correct

47. A variable is declared as:

```
DECLARE
    v_salary employees.salary%TYPE;
BEGIN
```

Which of the following is a correct use of the INTO clause?

Mark for Review
(1) Points

```
SELECT salary
INTO v_salary
FROM employees
WHERE employee_id=100;
(*)
```

```
SELECT v_salary
INTO salary
FROM employees
WHERE employee_id=100;
```

```
SELECT salary
FROM employees
INTO v_salary;
```

```
SELECT salary
FROM employees
WHERE employee_id=100
INTO v_salary;
```

Incorrect. Refer to Section 3.

48. A PL/SQL block includes the following statement:

```
SELECT last_name INTO v_last_name
FROM employees
WHERE employee_id=100;
```

What is the value of SQL%ISOPEN immediately after the SELECT statement is executed?

Mark for Review
(1) Points

True

False (*)

Null

Error. That attribute does not apply for implicit cursors.

Correct

49. There are no employees in Department 77. What will happen when the following block is executed?

```
BEGIN
DELETE FROM employees
WHERE department_id=77;
DBMS_OUTPUT.PUT_LINE(SQL%ROWCOUNT)
END;
```

Mark for Review
(1) Points

A NO_DATA_FOUND exception is raised.

A NULL is displayed.

A zero (0) is displayed. (*)

An exception is raised because the block does not contain a COMMIT statement.

Incorrect. Refer to Section 3.

50. Assume there are 5 employees in Department 10. What happens when the following statement is executed?

```
UPDATE employees
SET salary=salary*1.1;
```

Mark for Review
(1) Points

All employees get a 10% salary increase. (*)

No rows are modified because you did not specify "WHERE department_id=10"

A TOO_MANY_ROWS exception is raised.

An error message is displayed because you must use the INTO clause to hold

the new salary.

Correct

1. The main subject areas taught by the Oracle Academy are: Mark for Review
(1) Points

Computer Repairs

Database performance tuning.

Data Modeling, SQL and PL/SQL (*)

Systems programming and computer architecture

Correct

2. Users with detailed business knowledge can provide input to the Database Development Process? True or False? Mark for Review
(1) Points

True (*)

False

Correct

3. Software cannot operate without Hardware. True or False? Mark for Review
(1) Points

True (*)

False

Incorrect. Refer to Section 1

4. Databases were invented in 1989? True or False? Mark for Review
(1) Points

True

False (*)

Correct

5. Consider your school library. It will have a database with transaction details of which student borrows which books. Is a record of one student borrowing one book Data or Information? Mark for Review
(1) Points

Data (*)

Information

Both

Neither

Correct

6. What is the difference between "information" and "data"? Mark for Review
(1) Points

Data is held and understood only by users.

Information is held and understood only by users.

Information and data have no differences - they are two words for the same thing.

Data turns into information when it is stored in a database and accessed by systems and users. (*)

Correct

Section 2

7. Relationship names are usually verbs. True or False? Mark for Review
(1) Points

True (*)

False

Correct

8. All of the following could be attributes of an ENTITY called PERSON except one. Select the incorrect one: Mark for Review
(1) Points

Haircolor

Weight

Gender

Natacha Hansen (*)

Correct

9. Which of the following entities most likely contains valid attributes? (Choose two) Mark for Review
(1) Points

(Choose all correct answers)

Entity: Home. Attributes: Number of Bedrooms, Owner, Address, Date Built (*)

Entity: Pet. Attributes: Name, Birthdate, Owner (*)

Entity: Car. Attributes: Owner Occupation, Owner Salary, Speed

Entity: Mother. Attributes: Name, Birthdate, Occupation, Salary

Incorrect. Refer to Section 2

10. Primary Unique Identifiers: (Choose Two) Mark for Review
(1) Points

(Choose all correct answers)

Are required. The data model is not complete until all entities have a Primary UID. (*)

Distinguish each instance of an entity from all others. (*)

Are not required.

Can be created multiple times for an entity.

Correct