

Test: Final Exam Semester 1

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

Section 6

1. The following code does not violate any constraints and will not raise an ORA-02292 error. What will happen when the code is executed?

```
BEGIN
  DECLARE
    e_constraint_violation EXCEPTION;
    PRAGMA EXCEPTION_INIT(e_constraint_violation, -2292);
  BEGIN
    DBMS_OUTPUT.PUT_LINE('Inner block message');
  END;
EXCEPTION
  WHEN e_constraint_violation THEN
    DBMS_OUTPUT.PUT_LINE('Outer block message');
END;
```

Mark for Review
(1) Points

Inner block message' will be displayed.

The code will fail because the exception is declared in the inner block but is referenced in the outer block. (*)

Outer block message' will be displayed.

The code will fail because line 4 should read: PRAGMA EXCEPTION_INIT(-2292, e_constraint_violation);

Incorrect. Refer to Section 6.

2. Using two nested blocks, a TOO_MANY_ROWS exception is raised within the inner block. Which of the following exception handlers will successfully handle the exception? Mark for Review

(1) Points

WHEN TOO_MANY_ROWS in the inner block

WHEN TOO_MANY_ROWS in either block

WHEN OTHERS in either block

WHEN OTHERS in the inner block

All of the above (*)

Incorrect. Refer to Section 6.

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

```
<<outer>>
DECLARE
  v_myvar NUMBER;
```

PLSQL feedback final exam semester 1

```
BEGIN
  v_myvar := 10;
  DECLARE
    v_myvar NUMBER := 200;
  BEGIN
    outer.v_myvar := 20;
    v_myvar := v_myvar / 0; -- this raises a ZERO_DIVIDE error
    outer.v_myvar := 30;
  END;
  v_myvar := 40;
EXCEPTION
  WHEN ZERO_DIVIDE THEN
    DBMS_OUTPUT.PUT_LINE(v_myvar);
END;
```

Mark for Review
(1) Points

10

20 (*)

30

40

200

Incorrect. Refer to Section 6.

4. Using nested blocks, when is it necessary to label the outer block?. Mark for Review
(1) Points

You must always label the outer block.

You must always label both blocks.

You must label the outer block when two variables with the same name are declared, one in each block.

You must label the outer block when two variables with the same name are declared and you need to reference the outer block's variable within the inner block. (*)

Block labels are just comments and are therefore recommended but never needed.

Correct

5. There are no employees in department 75. what will be displayed when this code is executed?

```
DECLARE
  v_last_name employees.last_name%TYPE;
BEGIN
  DBMS_OUTPUT.PUT_LINE('A');
  BEGIN
    SELECT last_name INTO v_last_name
      FROM employees WHERE department_id = 75;
    DBMS_OUTPUT.PUT_LINE('B');
```

PLSQL feedback final exam semester 1

```
END;  
DBMS_OUTPUT.PUT_LINE('C');  
EXCEPTION  
  WHEN OTHERS THEN  
    DBMS_OUTPUT.PUT_LINE('D');  
END;
```

Mark for Review
(1) Points

A
C
D

A
D
(*)

A

A
B
D

None of the above

Incorrect. Refer to Section 6.

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

```
BEGIN -- outer block  
  DECLARE -- inner block  
    CURSOR emp_curs IS SELECT * FROM employees;  
    v_emp_rec emp_curs%ROWTYPE;  
  BEGIN  
    OPEN emp_curs;  
    LOOP  
      FETCH emp_curs INTO v_emp_rec;  
      DBMS_OUTPUT.PUT_LINE(v_emp_rec.salary);  
    END LOOP;  
  END;  
  CLOSE emp_curs;  
END;
```

Mark for Review
(1) Points

The code will fail because you cannot declare a cursor in an inner block.

The code will fail because the cursor is declared in the inner block but is referenced in the outer block. (*)

The code will execute successfully and display all the employees' salaries.

The code will execute forever because there is no statement to EXIT from the loop.

PLSQL feedback final exam semester 1

Correct

7. Which of the following best describes a user-defined exception? Mark for Review
(1) Points

A predefined Oracle Server error such as NO_DATA_FOUND

A non-predefined Oracle Server error such as ORA-01400

An error which is not automatically raised by the Oracle server (*)

Any error which has an Oracle error number of the form ORA-nnnnn

Correct

8. Which of the following are examples of predefined Oracle Server errors? (Choose three.) Mark for Review
(1) Points

(Choose all correct answers)

TOO_MANY_ROWS (*)

NO_DATA_FOUND (*)

OTHERS

ZERO_DIVIDE (*)

E_INSERT_EXCEP

Incorrect. Refer to Section 6.

9. An attempt to insert a null value into a NOT NULL table column raises an ORA-01400 exception. How can you code an exception handler to trap this exception? Mark for Review
(1) Points

Test for WHEN ORA-1400 in the exception section.

Declare a variable e_null_excep of type EXCEPTION, associate it with ORA-01400 using a PRAGMA directive, and test for WHEN e_null_excep in the exception section. (*)

Declare a variable e_null_excep of type VARCHAR2, associate it with ORA-01400 using a PRAGMA directive, and test for WHEN e_null_excep in the exception section.

Declare a variable as follows: e_null_excep EXCEPTION := -01400; Then test for WHEN e_null_excep in the exception section.

Correct

10. Which kinds of exceptions are raised implicitly (i.e., automatically)? (Choose two.) Mark for Review

(1) Points

(Choose all correct answers)

Predefined Oracle Server errors such as NO_DATA_FOUND (*)

User-defined errors

All errors

Non-predefined Oracle Server errors such as ORA-01400 (*)

Correct

Page 1 of 5

Test: Final Exam Semester 1

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

Section 6

11. How can you retrieve the error code and error message of any Oracle Server exception? Mark for Review

(1) Points

By using the functions SQLCODE and SQLERRM (*)

By using the functions SQLCODE and SQLERR

By using RAISE_APPLICATION_ERROR

By defining an EXCEPTION variable and using PRAGMA EXCEPTION_INIT

Incorrect. Refer to Section 6.

12. Examine the following code. What message or messages will be displayed when this code is executed?

```
DECLARE
  v_last_name employees.last_name%TYPE;
  v_number NUMBER := 27;
BEGIN
  v_number := v_number / 0;
  SELECT last_name INTO v_last_name FROM employees
    WHERE employee_id = 999;
EXCEPTION
  WHEN NO_DATA_FOUND THEN
    DBMS_OUTPUT.PUT_LINE('No rows were found');
  WHEN ZERO_DIVIDE THEN
    DBMS_OUTPUT.PUT_LINE('Attempt to divide by zero');
```

PLSQL feedback final exam semester 1

```
WHEN OTHERS THEN
    DBMS_OUTPUT.PUT_LINE('An error occurred');
END;
```

Mark for Review
(1) Points

- No rows were found
- Attempt to divide by zero (*)
- Attempt to divide by zero No rows were found
- An error occurred
- No message will be displayed

Incorrect. Refer to Section 6.

13. Which of the following is NOT an advantage of including an exception handler in a PL/SQL block? Mark for Review
(1) Points

- Protects the database from errors
- Code is more readable because error-handling routines can be written in the same block in which the error occurred
- Prevents errors from occurring (*)
- Avoids costly and time-consuming correction of mistakes

Correct

14. Which of the following are good practice guidelines for exception handling? (Choose three.) Mark for Review
(1) Points

(Choose all correct answers)

- Test your code with different combinations of data to see what potential errors can happen. (*)
- Use an exception handler whenever there is any possibility of an error occurring. (*)
- Include a WHEN OTHERS handler as the first handler in the exception section.
- Allow exceptions to propagate back to the calling environment.
- Handle specific named exceptions where possible, instead of relying on WHEN OTHERS. (*)

Incorrect. Refer to Section 6.

15. Which of the following best describes a PL/SQL exception? Mark for Review
(1) Points

PLSQL feedback final exam semester 1

A user enters an invalid password while trying to log on to the database.

An error occurs during execution which disrupts the normal operation of the program. (*)

A DML statement does not modify any rows.

The programmer makes a spelling mistake while writing the PL/SQL code.

Incorrect. Refer to Section 6.

16. Which of the following EXCEPTION sections are constructed correctly? (Choose two.) Mark for Review
(1) Points

(Choose all correct answers)

```
EXCEPTION
WHEN NO_DATA_FOUND THEN statement_1;
WHEN OTHERS THEN statement_2;
END;
```

(*)

```
EXCEPTION
WHEN OTHERS THEN statement_2;
WHEN NO_DATA_FOUND THEN statement_1;
END;
```

```
EXCEPTION
WHEN NO_DATA_FOUND THEN statement_1;
WHEN NO_DATA_FOUND THEN statement_2;
WHEN OTHERS THEN statement_3;
END;
```

```
EXCEPTION
WHEN OTHERS THEN statement_1;
END;
```

(*)

Incorrect. Refer to Section 6.

17. User-defined exceptions must be declared explicitly by the programmer, but then are raised automatically by the Oracle Server. True or False? Mark for Review
(1) Points

True

False (*)

Incorrect. Refer to Section 6.

18. There are no employees in department_id 99. What output will be displayed

when the following code is executed?

```
DECLARE
    v_count NUMBER;
BEGIN
    SELECT COUNT(*) INTO v_count
    FROM employees WHERE department_id = 99;
    IF v_count = 0 THEN
        RAISE NO_DATA_FOUND;
        DBMS_OUTPUT.PUT_LINE('No employees found');
    END IF;
EXCEPTION
    WHEN NO_DATA_FOUND THEN
        DBMS_OUTPUT.PUT_LINE('Department 99 is empty');
END;
```

Mark for Review
(1) Points

No employees found

No employees found Department 99 is empty

Department 99 is empty (*)

The block will fail because you cannot explicitly RAISE a predefined Oracle Server error such as NO_DATA_FOUND

Correct

19. Department-id 99 does not exist. what will be displayed when the following code is executed?

```
DECLARE
    v_deptname departments.department_name%TYPE;
BEGIN
    SELECT department_name INTO v_deptname
    FROM departments WHERE department_id = 99;
EXCEPTION
    WHEN NO_DATA_FOUND THEN
        RAISE_APPLICATION_ERROR(-20201,'Department does not exist');
END;
```

Mark for Review
(1) Points

ORA-01403: No Data Found ORA-20201: Department does not exist

ORA-01403: No Data Found

ORA-20201: Department does not exist (*)

None of the above

Incorrect. Refer to Section 6.

20. A user-defined exception can be raised:

- A. In the declaration section
- B. In the executable section
- C. In the exception section

Mark for Review
(1) Points

PLSQL feedback final exam semester 1

B

C

A and B

B and C (*)

A and C

Incorrect. Refer to Section 6.

Page 2 of 5

Test: Final Exam Semester 1

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

Section 7

21. What are the type of parameter modes? Mark for Review
(1) Points

CHARACTER, NUMBER, DATE, BOOLEAN

CONSTANT, VARIABLE, DEFAULT

LOCAL, GLOBAL, BOTH

IN, OUT, IN OUT (*)

Incorrect. Refer to Section 7.

22. Which parameter mode is the default? Mark for Review
(1) Points

IN (*)

OUT

NUMBER

VARIABLE

CONSTANT

Incorrect. Refer to Section 7.

PLSQL feedback final exam semester 1

23. Procedure SOMEPROC has five parameters named A, B, C, D, E in that order. The procedure was called as follows:
SOMEPROC(10,20,D=>50);

How was parameter D referenced?

Mark for Review

(1) Points

Positionally

Named (*)

A combination of positionally and named

A combination of named and defaulted

Defaulted

Incorrect. Refer to Section 7.

24. Procedure SOMEPROC has five parameters named A, B, C, D, E in that order. The procedure was called as follows:
SOMEPROC(10,20,D=>50);

How was parameter B referenced?

Mark for Review

(1) Points

Positional (*)

Named

A combination of positionally and named

A combination of named and defaulted

Defaulted

Correct

25. Which of the following can NOT be used as the datatype of a procedure parameter? Mark for Review

(1) Points

A non-SQL datatype such as BOOLEAN

The name of another procedure (*)

A large object datatype such as CLOB

A PLSQL record defined using %ROWTYPE

Incorrect. Refer to Section 7.

26. Examine the following procedure:
CREATE OR REPLACE PROCEDURE smallproc

```
(p_param IN NUMBER)
```

```
IS
```

```
BEGIN ....
```

The procedure is invoked by:

```
DECLARE
```

```
v_param NUMBER := 20;
```

```
BEGIN
```

```
smallproc(v_param);
```

```
END;
```

Which of the following statements is true? Mark for Review

(1) Points

p_param is a parameter and v_param is an argument

p_param is a formal parameter and 20 is an actual parameter

p_param is a formal parameter and v_param is an actual parameter (*)

p_param and v_param are both formal parameters, while 20 is an actual parameter

p_param is an actual parameter and v_param is a formal parameter

Correct

27. You want to create a procedure named SOMEPROC which accepts a single parameter named SOMEARM. The parameter can be up to 100 characters long. Which of the following is correct syntax to do this? Mark for Review

(1) Points

```
CREATE PROCEDURE someproc
(someparm varchar2)
```

```
IS
```

```
BEGIN ...
```

```
(*)
```

```
CREATE PROCEDURE someproc
(someparm varchar2(100) )
```

```
IS
```

```
BEGIN...
```

```
CREATE PROCEDURE someproc
```

```
IS
```

```
(someparm VARCHAR2)
```

```
BEGIN...
```

```
CREATE PROCEDURE someproc
someparm varchar2(100);
```

```
IS
```

```
BEGIN...
```

```
CREATE PROCEDURE someproc
(someparm 100)
```

```
IS
```

```
BEGIN ...
```

Incorrect. Refer to Section 7.

28. Which of the following is NOT correct coding for a procedure parameter? Mark for Review
(1) Points

- (p_param IN VARCHAR2)
- (p_param VARCHAR2)
- (p_param VARCHAR2(50)) (*)
- (p_param employees.last_name%TYPE)
- (p_param IN OUT VARCHAR2)

Incorrect. Refer to Section 7.

29. Which of the following statements about actual parameters is NOT true? Mark for Review
(1) Points

An actual parameter is declared in the calling environment, not in the called procedure

An actual parameter must be the name of a variable (*)

An actual parameter can have a Boolean datatype

The datatypes of an actual parameter and its formal parameter must be compatible

An actual parameter can have a TIMESTAMP datatype

Incorrect. Refer to Section 7.

30. You have created the following procedure:

```
CREATE OR REPLACE PROCEDURE double_it  
(p_param IN OUT NUMBER)  
IS
```

```
BEGIN  
p_param := p_param * 2;  
END;
```

Which of the following anonymous blocks invokes this procedure successfully? Mark for Review
(1) Points

```
BEGIN  
EXECUTE double_it(20);  
END;
```

```
BEGIN  
SELECT double_it(20)  
FROM DUAL;  
END;
```

```
DECLARE  
v_result NUMBER(6);  
BEGIN
```

```
v_result := double_it(20);
END;
```

```
DECLARE
v_result NUMBER(6) := 20;
BEGIN
double_it(v_result);
END; (*)
```

```
BEGIN
double_it(20);
END;
```

Incorrect. Refer to Section 7.

Page 3 of 5

Test: Final Exam Semester 1

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

Section 7

31. A programmer creates a PL/SQL subprogram which is compiled and stored in the database. Two separate users then execute an application which invokes this subprogram four times. How many times must the subprogram be recompiled? Mark for Review

(1) Points

Twice

Four times

None (*)

Eight times

Once

Incorrect. Refer to Section 7.

32. A programmer wants to create a PL/SQL procedure named EMP_PROC. what will happen when the following code is executed?

```
CREATE OR REPLACE PROCEDURE emp_proc IS
v_salary employees.salary%TYPE;
BEGIN
SELECT salary INTO v_salary FROM employees
WHERE employee_id = 999;
DBMS_OUTPUT.PUT_LINE('The salary is: ' || v_salary);
END;
```

PLSQL feedback final exam semester 1

Mark for Review
(1) Points

The statement will raise a NO_DATA_FOUND exception because employee_id 999 does not exist.

The statement will fail because the last line of code should be END emp_proc;

The statement will fail because you cannot declare variables such as v_salary inside a procedure.

The procedure will be created successfully. (*)

The statement will fail because the procedure does not have any parameters.

Incorrect. Refer to Section 7.

33. Which of the following keywords MUST be included in every PL/SQL procedure definition? (Choose three.) Mark for Review
(1) Points

(Choose all correct answers)

REPLACE

BEGIN (*)

IS or AS (*)

DECLARE

END (*)

Incorrect. Refer to Section 7.

34. Which of the following are characteristics of PL/SQL subprograms but not of anonymous PL/SQL blocks? (Choose three.) Mark for Review
(1) Points

(Choose all correct answers)

Can take parameters (*)

Are stored in the database (*)

Can begin with the keyword DECLARE

Are named (*)

Are compiled every time they are executed

Incorrect. Refer to Section 7.

35. Which of the following are benefits of using PL/SQL subprograms rather than anonymous blocks? (Choose three.) Mark for Review
(1) Points

PLSQL feedback final exam semester 1

(Choose all correct answers)

Easier to write

Better data security (*)

Easier code maintenance (*)

Faster performance (*)

Do not need to declare variables

Incorrect. Refer to Section 7.

36. The following are the steps involved in creating, and later modifying and re-creating, a PL/SQL procedure in Application Express. In what sequence should these steps be performed?

Retrieve the saved code from "Saved SQL" in SQL Commands

Execute the code to create the procedure

Execute the code to re-create the procedure

Click on the "Save" button and save the procedure code

Modify the code in the SQL Commands window

Type the procedure code in the SQL Commands window

Mark for Review

(1) Points

F,C,A,B,E,D

F,B,D,A,E,C (*)

E,D,F,C,A,B

F,B,D,E,A,C

F,B,C,D,E,A

Correct

Section 8

37. How do you specify that you want a procedure MYPROCA to use Invoker's Rights?

Mark for Review

(1) Points

CREATE OR REPLACE PROCEDURE myproca

AUTHID CURRENT_USER IS...

(*)

Invoker's Rights are the default, therefore no extra code is needed.

GRANT INVOKER TO myproca;

ALTER PROCEDURE myproca TO INVOKER;

PLSQL feedback final exam semister 1

CREATE OR REPLACE PROCEDURE myproca
AUTHID OWNER IS...

Incorrect. Refer to Section 8.

38. User REYHAN creates the following procedure: CREATE PROCEDURE proc1 AUTHID CURRENT_USER IS v_count NUMBER; BEGIN SELECT COUNT(*) INTO v_count FROM tom.employees; END; User BILL wants to execute this procedure. What privileges will BILL need? Mark for Review
(1) Points

EXECUTE on REYHAN.PROC1 and SELECT on TOM.EMPLOYEES (*)

EXECUTE on REYHAN.PROC1

SELECT on TOM.EMPLOYEES

BILL needs no privileges

None of the above. The procedure will fail to compile because REYHAN does not have SELECT privilege on TOM.EMPLOYEES.

Incorrect. Refer to Section 8.

39. You have created a function named IS_LEAPYEAR that accepts one IN parameter of datatype DATE and returns a Boolean value (TRUE or FALSE) depending on whether the date is in a leap year. What is wrong with this query:
SELECT last_name, hire_date
FROM employees
WHERE is_leapyear(hire_date)=TRUE;

Mark for Review
(1) Points

The IS_LEAPYEAR function must be in the SELECT clause, not the WHERE clause.

You cannot use DATE and BOOLEAN datatypes in the same function.

The SELECT statement returns more than one row.

IS_LEAPYEAR is a reserved word in the SQL language.

The function returns a Boolean, and therefore cannot be used within a SELECT statement. (*)

Incorrect. Refer to Section 8.

40. Where can a function be used in a query? Mark for Review
(1) Points

Nowhere in a query.

Anywhere in a query. (*)

Only in the SELECT clause

Only in the WHERE clause

In the SELECT or WHERE clauses, but not in the ORDER BY clause.

Incorrect. Refer to Section 8.

Page 4 of 5

Test: Final Exam Semester 1

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

Section 8

41. In which DML statements can user-defined functions be used? Mark for Review
(1) Points

INSERT and UPDATE, but not DELETE.

INSERT only.

All DML statements. (*)

UPDATE only

DELETE only

Correct

42. A function must have at least one IN parameter, and must return exactly one value. Mark for Review
(1) Points

True

False (*)

Correct

43. You have created a function named NEWFUNC. You now change some of the function code, and try to recreate the function by executing:
CREATE OR REPLACE FUNCTION newfunc ;
what happens?
Mark for Review
(1) Points

The command fails because the function already exists.

The function is automatically dropped and then recreated. (*)

The command fails because you should execute: CREATE AND REPLACE;

A second function named NEWFUNC_2 is created.

The function is dropped but not recreated.

Correct

44. A function named MYFUNC has been created. This function accepts one IN parameter of datatype VARCHAR2 and returns a NUMBER. You want to invoke the function within the following anonymous block:

```
DECLARE
```

```
v_var1 NUMBER(6,2);
```

```
BEGIN
```

```
-- Line A
```

```
END;
```

What could be coded at Line A?

Mark for Review

(1) Points

```
myfunc('Crocodile') := v_var1;
```

```
myfunc(v_var1) := 'Crocodile';
```

```
myfunc(v_var1, 'Crocodile');
```

```
v_var1 := myfunc('Crocodile'); (*)
```

```
myfunc('Crocodile', v_var1);
```

Incorrect. Refer to Section 8.

45. What is wrong with the following code?

```
CREATE FUNCTION badfunc
```

```
(p_param NUMBER(4))
```

```
RETURN BOOLEAN
```

```
IS BEGIN
```

```
RETURN (p_param > 10);
```

```
END badfunc;
```

Mark for Review

(1) Points

P_PARAM must be declared AFTER the RETURN clause.

P_PARAM must have a default value.

The datatype of the IN parameter cannot have a precision or scale. It must be NUMBER, not NUMBER(4). (*)

RETURN (p_param > 10); is wrong because you cannot return an expression.

The NUMBER datatype must have a scale as well as a precision.

Incorrect. Refer to Section 8.

PLSQL feedback final exam semester 1

46. Why will this function not compile correctly?

```
CREATE FUNCTION bad_one  
IS BEGIN  
    RETURN NULL;  
END bad_one;
```

Mark for Review
(1) Points

You cannot RETURN a NULL.

You must declare the type of the RETURN before the IS. (*)

You must have at least one IN parameter.

You must code CREATE OR REPLACE, not CREATE.

The body of the function must contain at least one executable statement (as well as RETURN).

Incorrect. Refer to Section 8.

47. Which of the following is a difference between a procedure and a function?

Mark for Review
(1) Points

Functions cannot be nested; procedures can be nested to at least 8 levels.

A procedure can have default values for parameters, while a function cannot.

An explicit cursor can be declared in a procedure, but not in a function.

A function cannot be used within a SQL statement; a procedure can be used within SQL.

A function must return a value, a procedure may or may not. (*)

Incorrect. Refer to Section 8.

48. The following code shows the dependencies between three procedures:

```
CREATE PROCEDURE parent  
IS BEGIN  
    child1;  
    child2;  
END parent;  
You now try to execute:
```

```
DROP PROCEDURE child2;  
What happens?
```

Mark for Review
(1) Points

You cannot drop CHILD2 because PARENT is dependent on it.

CHILD2 is dropped successfully. PARENT and CHILD1 are both marked INVALID.

The database automatically drops PARENT as well.

PLSQL feedback final exam semester 1

CHILD2 is dropped successfully. PARENT is marked INVALID. CHILD1 is still valid.
(*)

The database automatically drops CHILD1 as well.

Incorrect. Refer to Section 8.

49. Examine the following code (the code of CHILD2 is not shown):

```
CREATE PROCEDURE child1
IS v_salary employees.salary%TYPE;
BEGIN
SELECT salary INTO v_salary FROM employees
WHERE employee_id = 9999;
EXCEPTION
WHEN NO_DATA_FOUND THEN NULL;
END child1;
CREATE PROCEDURE parent
```

```
IS BEGIN
child1;
child2;
EXCEPTION
WHEN NO_DATA_FOUND THEN NULL;
END parent;
```

Employee_id 9999 does not exist. What happens when PARENT is executed?

Mark for Review

(1) Points

CHILD1 handles the exception successfully and ends. PARENT continues to execute and invokes CHILD2. (*)

CHILD1 ends abruptly, PARENT handles the exception successfully and ends. CHILD2 does not execute.

CHILD1 ends abruptly, then PARENT also ends abruptly with an unhandled exception.

PARENT handles the exception, then CHILD1 resumes execution.

PARENT fails to compile because you cannot have the same exception handler in two separate subprograms.

Incorrect. Refer to Section 8.

50. You want to remove the procedure NO_NEED from your schema. You execute:

```
DROP PROCEDURE no_need;
```

Which Data Dictionary views are updated automatically?

Mark for Review

(1) Points

USER_PROCEEDURES

USER_OBJECTS

USER_SOURCE

All of the above. (*)

None of the above.

Correct

Page 5 of 5

8. 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;
```

High Paid

Low Paid

Null

The code will fail and return an exception (*)

19. 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?

NULL

won'

lost' (*)

False

23. 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?

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 (*)

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;

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.

8. 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;

High Paid

Low Paid

Null

The code will fail and return an exception (*)

19. 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?

NULL

won'

lost' (*)

False

23. 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?

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 (*)

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;
```

20 job_ids will be displayed.

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

PLSQL feedback final exam semister 1

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.

8. 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;
```

High Paid

Low Paid

Null

The code will fail and return an exception (*)

19. 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?

NULL

won'

lost' (*)

False

23. 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?

WHILE NOT v_boolean LOOP

PLSQL feedback final exam semester 1

```
WHILE v_boolean AND v_counter < 6 LOOP
```

```
WHILE v_counter > 8 LOOP
```

```
WHILE v_counter IN 1..5 LOOP (*)
```

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;
```

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.

Test: Final Exam Semester 1

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

Section 6

1. There are no employees in department 75. what will be displayed when this code is executed?

```
DECLARE
  v_last_name employees.last_name%TYPE;
BEGIN
  DBMS_OUTPUT.PUT_LINE('A');
  BEGIN
    SELECT last_name INTO v_last_name
      FROM employees WHERE department_id = 75;
    DBMS_OUTPUT.PUT_LINE('B');
  END;
  DBMS_OUTPUT.PUT_LINE('C');
EXCEPTION
  WHEN OTHERS THEN
    DBMS_OUTPUT.PUT_LINE('D');
END;
```

PLSQL feedback final exam semester 1

Mark for Review
(1) Points

A
C
D

A
D
(*)

A

A
B
D

None of the above

Incorrect. Refer to Section 6.

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

```
<<outer>>
DECLARE
    v_myvar NUMBER;
BEGIN
    v_myvar := 10;
    DECLARE
        v_myvar NUMBER := 200;
    BEGIN
        outer.v_myvar := 20;
        v_myvar := v_myvar / 0; -- this raises a ZERO_DIVIDE error
        outer.v_myvar := 30;
    END;
    v_myvar := 40;
EXCEPTION
    WHEN ZERO_DIVIDE THEN
        DBMS_OUTPUT.PUT_LINE(v_myvar);
END;
```

Mark for Review
(1) Points

10
20 (*)
30
40
200

Incorrect. Refer to Section 6.

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

```
BEGIN -- outer block
  DECLARE -- inner block
    CURSOR emp_curs IS SELECT * FROM employees;
    v_emp_rec emp_curs%ROWTYPE;
  BEGIN
    OPEN emp_curs;
    LOOP
      FETCH emp_curs INTO v_emp_rec;
      DBMS_OUTPUT.PUT_LINE(v_emp_rec.salary);
    END LOOP;
  END;
CLOSE emp_curs;
END;
```

Mark for Review
(1) Points

The code will fail because you cannot declare a cursor in an inner block.

The code will fail because the cursor is declared in the inner block but is referenced in the outer block. (*)

The code will execute successfully and display all the employees' salaries.

The code will execute forever because there is no statement to EXIT from the loop.

Incorrect. Refer to Section 6.

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

```
DECLARE
  e_excep1 EXCEPTION;
  e_excep2 EXCEPTION;
BEGIN
  RAISE e_excep1;
EXCEPTION
  WHEN e_excep1 THEN BEGIN
    RAISE e_excep2; END;
END;
```

Mark for Review
(1) Points

It will fail to compile because you cannot have a subblock inside an exception section.

It will fail to compile because e_excep1 is out of scope in the subblock.

It will fail to compile because you cannot declare more than one exception in the same block.

It will compile successfully and return an unhandled e_excep2 to the calling environment. (*)

Incorrect. Refer to Section 6.

5. Examine the following code which shows three levels of nested block. what is the scope of the variable v_middle_var?

```
DECLARE -- outer block
    v_outer_var NUMBER;
BEGIN
    DECLARE -- middle block
        v_middle_var NUMBER;
    BEGIN
        DECLARE -- inner block
            v_inner_var NUMBER;
        BEGIN
            ...
        END;
    END;
END;
```

Mark for Review
(1) Points

- All three blocks
- Middle and outer blocks only
- Middle and inner blocks only (*)
- Middle block only
- None of the above

Incorrect. Refer to Section 6.

6. what will happen when the following code is executed?

```
DECLARE
    e_outer_excep EXCEPTION;
BEGIN
    DECLARE
        e_inner_excep EXCEPTION;
    BEGIN
        RAISE e_outer_excep;
    END;
EXCEPTION
    WHEN e_outer_excep THEN
        DBMS_OUTPUT.PUT_LINE('Outer raised');
    WHEN e_inner_excep THEN
        DBMS_OUTPUT.PUT_LINE('Inner raised');
END;
```

Mark for Review
(1) Points

The code will fail to compile because e_inner_excep cannot be referenced in the outer block. (*)

The code will propagate the e_outer_excep back to the calling environment.

The code will execute successfully and 'Outer Raised' will be displayed.

The code will fail to compile because e_inner_excep was declared but never RAISED.

PLSQL feedback final exam semester 1
Incorrect. Refer to Section 6.

7. An attempt to update an employee's salary to a negative value will violate a check constraint and raise an ORA-02290 exception. Which of the following is a correct definition of a handler for this exception? Mark for Review
(1) Points

```
DECLARE
e_sal_excep EXCEPTION;
PRAGMA EXCEPTION_INIT(-02290,e_sal_excep);
```

```
DECLARE
PRAGMA EXCEPTION_INIT(e_sal_excep,-02290);
e_sal_excep EXCEPTION;
```

```
DECLARE
e_sal_excep EXCEPTION;
PRAGMA EXCEPTION_INIT(e_sal_excep,-02290);
(*)
```

```
DECLARE
e_sal_excep EXCEPTION;
PRAGMA_EXCEPTION_INIT(e_sal_exception,-02290);
```

```
DECLARE
e_sal_excep EXCEPTION;
PRAGMA EXCEPTION_INIT(e_sal_excep,02290);
```

Incorrect. Refer to Section 6.

8. Examine the following code. Which exception handlers would successfully trap the exception which will be raised when this code is executed? (Choose two.)

```
DECLARE
    CURSOR emp_curs IS SELECT * FROM employees;
    v_emp_rec emp_curs%ROWTYPE;
BEGIN
    FETCH emp_curs INTO v_emp_rec;
    OPEN emp_curs;
    CLOSE emp_curs;
EXCEPTION ...
END;
```

Mark for Review
(1) Points

(Choose all correct answers)

WHEN CURSOR_NOT_OPEN

WHEN INVALID_CURSOR (*)

WHEN OTHERS (*)

WHEN NO_DATA_FOUND

WHEN INVALID_FETCH

Incorrect. Refer to Section 6.

9. Which of the following best describes a predefined Oracle Server error? Mark for Review
(1) Points

Has a standard Oracle error number but must be named by the PL/SQL programmer

Is not raised automatically but must be declared and raised explicitly by the PL/SQL programmer

Has a standard Oracle error number and a standard name which can be referenced in the EXCEPTION section (*)

Is associated with an Oracle error number using PRAGMA EXCEPTION_INIT

Incorrect. Refer to Section 6.

10. Which of the following are examples of predefined Oracle Server errors? (Choose three.) Mark for Review
(1) Points

(Choose all correct answers)

TOO_MANY_ROWS (*)

NO_DATA_FOUND (*)

OTHERS

ZERO_DIVIDE (*)

E_INSERT_EXCEP

Incorrect. Refer to Section 6.

Page 1 of 5

Test: Final Exam Semester 1

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

Section 6

11. An attempt to insert a null value into a NOT NULL table column raises an ORA-01400 exception. How can you code an exception handler to trap this exception?

Mark for Review
(1) Points

Test for WHEN ORA-1400 in the exception section.

Declare a variable e_null_excep of type EXCEPTION, associate it with ORA-01400 using a PRAGMA directive, and test for WHEN e_null_excep in the exception section. (*)

Declare a variable e_null_excep of type VARCHAR2, associate it with ORA-01400 using a PRAGMA directive, and test for WHEN e_null_excep in the exception section.

Declare a variable as follows: e_null_excep EXCEPTION := -01400; Then test for WHEN e_null_excep in the exception section.

Correct

12. Examine the following code fragment. At Line A, you want to raise an exception if the fetched salary value is greater than 30000. How can you do this?

```
DECLARE
  v_salary employees.salary%TYPE;
BEGIN
  SELECT salary INTO v_salary FROM employees
    WHERE employee_id = 100;
  IF v_salary > 30000 THEN
    -- Line A
  END IF;
...

```

Mark for Review
(1) Points

Test for WHEN VALUE_TOO_HIGH in the exception section.

Use RAISE_APPLICATION_ERROR to raise an exception explicitly. (*)

Test for WHEN OTHERS in the exception section, because WHEN OTHERS traps all exceptions.

Define an EXCEPTION variable and associate it with an Oracle Server error number using PRAGMA EXCEPTION_INIT.

Correct

13. Department-id 99 does not exist. What will be displayed when the following code is executed?

```
DECLARE
  v_deptname departments.department_name%TYPE;
BEGIN
  SELECT department_name INTO v_deptname
    FROM departments WHERE department_id = 99;
EXCEPTION
  WHEN NO_DATA_FOUND THEN
    RAISE_APPLICATION_ERROR(-20201, 'Department does not exist');
END;

```

Mark for Review
(1) Points

ORA-01403: No Data Found ORA-20201: Department does not exist

PLSQL feedback final exam semester 1

ORA-01403: No Data Found

ORA-20201: Department does not exist (*)

None of the above

Incorrect. Refer to Section 6.

14. A user-defined exception must be declared as a variable of data type EXCEPTION. True or False? Mark for Review
(1) Points

True (*)

False

Correct

15. There are no employees in department_id 99. what output will be displayed when the following code is executed?

```
DECLARE
    v_count NUMBER;
BEGIN
    SELECT COUNT(*) INTO v_count
    FROM employees WHERE department_id = 99;
    IF v_count = 0 THEN
        RAISE NO_DATA_FOUND;
        DBMS_OUTPUT.PUT_LINE('No employees found');
    END IF;
EXCEPTION
    WHEN NO_DATA_FOUND THEN
        DBMS_OUTPUT.PUT_LINE('Department 99 is empty');
END;
```

Mark for Review
(1) Points

No employees found

No employees found Department 99 is empty

Department 99 is empty (*)

The block will fail because you cannot explicitly RAISE a predefined Oracle Server error such as NO_DATA_FOUND

Incorrect. Refer to Section 6.

16. There are no employees in department 99. what message or messages will be displayed when the following code is executed?

```
DECLARE
    e_my_excep EXCEPTION;
BEGIN
    BEGIN
        UPDATE employees SET salary = 10000
        WHERE department_id = 99;
```


PLSQL feedback final exam semister 1

```
IF SQL%ROWCOUNT = 0 THEN
    RAISE e_my_excep;
END IF;
EXCEPTION
    WHEN e_my_excep THEN
        DBMS_OUTPUT.PUT_LINE('Message 1');
        RAISE e_my_excep;
        DBMS_OUTPUT.PUT_LINE('Message 2');
END;
DBMS_OUTPUT.PUT_LINE('Message 3');
EXCEPTION
    WHEN e_my_excep THEN
        DBMS_OUTPUT.PUT_LINE('Message 4');
END;
```

Mark for Review
(1) Points

Message 1
Message 3

Message 1
Message 2

Message 1
Message 3
Message 4

Message 1
Message 4
(*)

Incorrect. Refer to Section 6.

17. which of the following EXCEPTION sections are constructed correctly? (Choose two.) Mark for Review
(1) Points

(Choose all correct answers)

```
EXCEPTION
    WHEN NO_DATA_FOUND THEN statement_1;
    WHEN OTHERS THEN statement_2;
END;
(*)
```

```
EXCEPTION
    WHEN OTHERS THEN statement_2;
    WHEN NO_DATA_FOUND THEN statement_1;
END;
```

```
EXCEPTION
    WHEN NO_DATA_FOUND THEN statement_1;
    WHEN NO_DATA_FOUND THEN statement_2;
    WHEN OTHERS THEN statement_3;
```

END;

```

    EXCEPTION
    WHEN OTHERS THEN statement_1;
END;
(*)

```

Incorrect. Refer to Section 6.

18. Which of the following are good practice guidelines for exception handling?
(Choose three.) Mark for Review
(1) Points

(Choose all correct answers)

Test your code with different combinations of data to see what potential errors can happen. (*)

Use an exception handler whenever there is any possibility of an error occurring. (*)

Include a WHEN OTHERS handler as the first handler in the exception section.

Allow exceptions to propagate back to the calling environment.

Handle specific named exceptions where possible, instead of relying on WHEN OTHERS. (*)

Incorrect. Refer to Section 6.

19. Examine the following code. Why does the exception handler not follow good practice guidelines?

```

DECLARE
    v_salary employees.salary%TYPE;
BEGIN
    SELECT salary INTO v_salary FROM employees
    WHERE employee_id = 999;
EXCEPTION
    WHEN OTHERS THEN
        DBMS_OUTPUT.PUT_LINE('An error occurred');
END;

```

Mark for Review
(1) Points

You should not use DBMS_OUTPUT.PUT_LINE in an exception handler.

employee_id 999 does not exist in the employees table.

The exception handler should test for the named exception NO_DATA_FOUND. (*)

The exception handler should COMMIT the transaction.

Incorrect. Refer to Section 6.

20. Which of the following is NOT an advantage of including an exception handler in a PL/SQL block? Mark for Review
(1) Points

Protects the database from errors

Code is more readable because error-handling routines can be written in the same block in which the error occurred

Prevents errors from occurring (*)

Avoids costly and time-consuming correction of mistakes

Incorrect. Refer to Section 6.

Page 2 of 5

Test: Final Exam Semester 1

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

Section 7

21. The following are the steps involved in creating, and later modifying and re-creating, a PL/SQL procedure in Application Express. In what sequence should these steps be performed?

Retrieve the saved code from "Saved SQL" in SQL Commands

Execute the code to create the procedure

Execute the code to re-create the procedure

Click on the "Save" button and save the procedure code

Modify the code in the SQL Commands window

Type the procedure code in the SQL Commands window

Mark for Review

(1) Points

F,C,A,B,E,D

F,B,D,A,E,C (*)

E,D,F,C,A,B

F,B,D,E,A,C

F,B,C,D,E,A

Incorrect. Refer to Section 7.

22. A programmer creates a PL/SQL subprogram which is compiled and stored in the database. Two separate users then execute an application which invokes this

subprogram four times. How many times must the subprogram be recompiled? Mark for Review

(1) Points

Twice

Four times

None (*)

Eight times

Once

Incorrect. Refer to Section 7.

23. A PL/SQL procedure named MYPROC has already been created and stored in the database. Which of the following will successfully re-create the procedure after some changes have been made to the code? Mark for Review

(1) Points

CREATE PROCEDURE myproc IS ...

CREATE OR REPLACE PROCEDURE myproc IS (*)

UPDATE PROCEDURE myproc IS ...

ALTER PROCEDURE myproc IS ...

None of the above, because the procedure must be dropped before it can be re-created.

Incorrect. Refer to Section 7.

24. Which of the following are benefits of using PL/SQL subprograms rather than anonymous blocks? (Choose three.) Mark for Review

(1) Points

(Choose all correct answers)

Easier to write

Better data security (*)

Easier code maintenance (*)

Faster performance (*)

Do not need to declare variables

Incorrect. Refer to Section 7.

25. A PL/SQL procedure named MY_PROC1 has been successfully created in the database. The procedure has no parameters. Which of the following will successfully invoke the procedure in Application Express? (Choose two.) Mark for Review

(1) Points

PLSQL feedback final exam semester 1

(Choose all correct answers)

```
DECLARE
  v_var1 NUMBER := 20;
BEGIN
  my_proc1(v_var1);
END;
```

```
EXECUTE my_proc1;
```

```
BEGIN
  my_proc1;
END;
(*)
```

```
CREATE OR REPLACE PROCEDURE my_proc2 IS
BEGIN
  my_proc1;
END my_proc2;
(*)
```

```
SELECT my_proc1 FROM DUAL;
```

Incorrect. Refer to Section 7.

26. Which of the following are characteristics of PL/SQL stored procedures?
(Choose three.) Mark for Review
(1) Points

(Choose all correct answers)

They are named PL/SQL blocks (*)

They must return exactly one value to the calling environment.

They can have an exception section. (*)

They can be invoked from inside a SQL statement.

They can accept parameters. (*)

Incorrect. Refer to Section 7.

27. Procedure SOMEPROC has five parameters named A, B, C, D, E in that order. The procedure was called as follows:
SOMEPROC(10,20,D=>50);

How was parameter D referenced?
Mark for Review
(1) Points

Positionally

Named (*)

PLSQL feedback final exam semester 1

A combination of positionally and named

A combination of named and defaulted

Defaulted

Incorrect. Refer to Section 7.

28. Which parameter mode is the default? Mark for Review
(1) Points

IN (*)

OUT

NUMBER

VARIABLE

CONSTANT

Incorrect. Refer to Section 7.

29. The following procedure has been created:
CREATE OR REPLACE PROCEDURE defproc
(A IN NUMBER := 50,
B IN NUMBER,
C IN NUMBER DEFAULT 40)
IS
Which one of the following will invoke the procedure correctly?
Mark for Review
(1) Points

defproc(30 => A);

defproc(30, 60 => C);

defproc(40, 70); (*)

defproc(10 => A, 25 => C);

defproc;

Correct

30. Procedure SOMEPROC has five parameters named A, B, C, D, E in that order. The procedure was called as follows:
SOMEPROC(10,20,D=>50);

How was parameter B referenced?
Mark for Review
(1) Points

Positional (*)

Named

PLSQL feedback final exam semester 1

A combination of positionally and named

A combination of named and defaulted

Defaulted

Correct

Page 3 of 5

Test: Final Exam Semester 1

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

Section 7

31. Which of the following best describes how an IN parameter affects a procedure? Mark for Review
(1) Points

It describes the order in which the procedure's statements should be executed.

It describes which parts of the procedure's code are optional or conditional.

It makes the procedure execute faster.

It passes a value into the procedure when the procedure is invoked. (*)

It allows complex calculations to be executed inside the procedure.

Incorrect. Refer to Section 7.

32. You have created the following procedure:

```
CREATE OR REPLACE PROCEDURE double_it  
(p_param IN OUT NUMBER)  
IS
```

```
BEGIN
```

```
p_param := p_param * 2;
```

```
END;
```

Which of the following anonymous blocks invokes this procedure successfully? Mark for Review
(1) Points

```
BEGIN
```

```
EXECUTE double_it(20);
```

```
END;
```

```
BEGIN
```

```
SELECT double_it(20)
```

```
FROM DUAL;
END;
```

```
DECLARE
v_result NUMBER(6);
BEGIN
v_result := double_it(20);
END;
```

```
DECLARE
v_result NUMBER(6) := 20;
BEGIN
double_it(v_result);
END; (*)
```

```
BEGIN
double_it(20);
END;
```

Incorrect. Refer to Section 7.

33. Examine the following procedure:

```
CREATE OR REPLACE PROCEDURE smallproc
(p_param IN NUMBER)
```

```
IS
```

```
BEGIN ....
```

The procedure is invoked by:

```
DECLARE
```

```
v_param NUMBER := 20;
```

```
BEGIN
```

```
smallproc(v_param);
```

```
END;
```

which of the following statements is true? Mark for Review

(1) Points

p_param is a parameter and v_param is an argument

p_param is a formal parameter and 20 is an actual parameter

p_param is a formal parameter and v_param is an actual parameter (*)

p_param and v_param are both formal parameters, while 20 is an actual parameter

p_param is an actual parameter and v_param is a formal parameter

Incorrect. Refer to Section 7.

34. which of the following is NOT correct coding for a procedure parameter? Mark for Review

(1) Points

(p_param IN VARCHAR2)

(p_param VARCHAR2)

(p_param VARCHAR2(50)) (*)

(p_param employees.last_name%TYPE)

PLSQL feedback final exam semister 1
(p_param IN OUT VARCHAR2)

Correct

35. You want to create a procedure named SOMEPROC which accepts a single parameter named SOMEPARAM. The parameter can be up to 100 characters long. Which of the following is correct syntax to do this? Mark for Review
(1) Points

```
CREATE PROCEDURE someproc  
(someparam varchar2)  
IS  
BEGIN ...  
(*)
```

```
CREATE PROCEDURE someproc  
(someparam varchar2(100) )  
IS  
BEGIN...
```

```
CREATE PROCEDURE someproc  
IS  
(someparam VARCHAR2)  
BEGIN...
```

```
CREATE PROCEDURE someproc  
someparam varchar2(100);  
IS  
BEGIN...
```

```
CREATE PROCEDURE someproc  
(someparam 100)  
IS  
BEGIN ...
```

Correct

36. You have created a procedure named MYPROC that accepts three IN parameters A, B, and C (all numbers). Which of the following calls to MYPROC is NOT correct? Mark for Review
(1) Points

```
myproc(5,10,20);  
myproc(a=>5,b=>10,20) (*)  
myproc(a=>5,b=>10,c=>20)  
myproc(5,10,c=>20)
```

Incorrect. Refer to Section 7.

Section 8

37. Which of the following is a difference between a procedure and a function?

Mark for Review

(1) Points

Functions cannot be nested; procedures can be nested to at least 8 levels.

A procedure can have default values for parameters, while a function cannot.

An explicit cursor can be declared in a procedure, but not in a function.

A function cannot be used within a SQL statement; a procedure can be used within SQL.

A function must return a value, a procedure may or may not. (*)

Incorrect. Refer to Section 8.

38. In a SELECT statement, where can a function NOT be used? Mark for Review

(1) Points

In a GROUP BY or HAVING clause.

A function can be used anywhere in a SELECT statement. (*)

In a WHERE clause.

In the column list (SELECT) clause.

In an ORDER BY clause.

Correct

39. You have created a function named NEWFUNC. You now change some of the function code, and try to recreate the function by executing:

```
CREATE OR REPLACE FUNCTION newfunc .... ;
```

What happens?

Mark for Review

(1) Points

The command fails because the function already exists.

The function is automatically dropped and then recreated. (*)

The command fails because you should execute: CREATE AND REPLACE;

A second function named NEWFUNC_2 is created.

The function is dropped but not recreated.

Incorrect. Refer to Section 8.

40. Examine the following code:

```
CREATE OR REPLACE FUNCTION add_func  
(p_param1 NUMBER, p_param2 NUMBER)
```

PLSQL feedback final exam semester 1

```
RETURN NUMBER
```

```
IS
```

```
BEGIN
```

```
RETURN (p_param1 + p_param2);
```

```
END;
```

What will be displayed when the following SQL statement is executed?

```
SELECT add_func(6, add_func(3,8)) FROM dual;
```

Mark for Review

(1) Points

23

11

66

17 (*)

An error message will be displayed because you cannot nest user-defined functions.

Incorrect. Refer to Section 8.

Page 4 of 5

Test: Final Exam Semester 1

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

Section 8

41. Which of the following is a difference between a procedure and a function?

Mark for Review

(1) Points

A procedure can include DML statements, but a function cannot.

A function must have at least one IN parameter, while parameters are optional for a procedure.

A procedure can return a BOOLEAN datatype, while a function cannot.

A function can be used inside a SQL statement, while a procedure cannot. (*)

A procedure can include an EXCEPTION section, while a function cannot.

Incorrect. Refer to Section 8.

42. You try to create a function named MYFUNC. The function does not compile correctly because there are errors in your code. Which Dictionary view can you query

to see the errors? Mark for Review

(1) Points

```
USER_SOURCE
USER_ERRORS (*)
USER_OBJECTS
USER_DEPENDENCIES
USER_COMPILES
```

Incorrect. Refer to Section 8.

43. Examine the following code (the code of CHILD2 is not shown):

```
CREATE PROCEDURE child1
IS v_salary employees.salary%TYPE;
BEGIN
SELECT salary INTO v_salary FROM employees
WHERE employee_id = 9999;
EXCEPTION
WHEN NO_DATA_FOUND THEN NULL;
END child1;
CREATE PROCEDURE parent
```

```
IS BEGIN
child1;
child2;
EXCEPTION
WHEN NO_DATA_FOUND THEN NULL;
END parent;
```

Employee_id 9999 does not exist. What happens when PARENT is executed?

Mark for Review

(1) Points

CHILD1 handles the exception successfully and ends. PARENT continues to execute and invokes CHILD2. (*)

CHILD1 ends abruptly, PARENT handles the exception successfully and ends. CHILD2 does not execute.

CHILD1 ends abruptly, then PARENT also ends abruptly with an unhandled exception.

PARENT handles the exception, then CHILD1 resumes execution.

PARENT fails to compile because you cannot have the same exception handler in two separate subprograms.

Correct

44. Which Data Dictionary view can be used to display the detailed code of a procedure in your schema? Mark for Review

(1) Points

```
USER_PROCEDURES
USER_OBJECTS
```

PLSQL feedback final exam semester 1

USER_SOURCE (*)

USER_SUBPROGRAMS

None of the above.

Correct

45. You want to see the names, modes and data types of the formal parameters of function MY_FUNC in your schema. How can you do this? (Choose two) Mark for Review
(1) Points

(Choose all correct answers)

Query USER_PARAMETERS

Query USER_SOURCE (*)

Query USER_FUNCTIONS

SHOW PARAMETER my_func;

DESCRIBE my_func; (*)

Incorrect. Refer to Section 8.

46. How do you specify that you want a procedure MYPROCA to use "Definer's Rights"? Mark for Review
(1) Points

CREATE OR REPLACE PROCEDURE myproca
AUTHID CURRENT_USER IS...

CREATE OR REPLACE PROCEDURE myproca
AUTHID OWNER IS...

GRANT DEFINER TO myproca;

ALTER PROCEDURE myproca TO DEFINER;

Definer's Rights are the default, therefore no extra code or commands are needed. (*)

Incorrect. Refer to Section 8.

47. User REYHAN creates the following procedure: CREATE PROCEDURE proc1 AUTHID CURRENT_USER IS v_count NUMBER; BEGIN SELECT COUNT(*) INTO v_count FROM tom.employees; END; User BILL wants to execute this procedure. What privileges will BILL need? Mark for Review
(1) Points

EXECUTE on REYHAN.PROC1 and SELECT on TOM.EMPLOYEES (*)

PLSQL feedback final exam semester 1

EXECUTE on REYHAN.PROC1

SELECT on TOM.EMPLOYEES

BILL needs no privileges

None of the above. The procedure will fail to compile because REYHAN does not have SELECT privilege on TOM.EMPLOYEES.

Incorrect. Refer to Section 8.

48. What is one of the advantages of using user-defined functions in a SQL statement? Mark for Review
(1) Points

They automate repetitive formulas which otherwise you would have to type in full every time you used them. (*)

They execute faster than system-defined functions such as UPPER and LOWER.

They allow you to execute DML from inside a SELECT statement.

They allow you to use functions which return a BOOLEAN.

They are stored on your local PC, not in the database.

Incorrect. Refer to Section 8.

49. Which of the following are NOT allowed in a function which is used inside a SQL statement which updates the EMPLOYEES table? (Choose two). Mark for Review
(1) Points

(Choose all correct answers)

SELECT FROM departments;

COMMIT; (*)

A RETURN statement.

DDL statements such as CREATE or ALTER. (*)

A WHEN OTHERS exception handler.

Incorrect. Refer to Section 8.

50. Which one of the following statements about user-defined functions is NOT true? Mark for Review
(1) Points

They can execute spell-checking routines.

They can be used inside SQL statements.

They can be combined (nested) together, similar to nesting system functions, for example INITCAP(SUBSTR(.....)).

PLSQL feedback final exam semester 1

They can return a TIMESTAMP datatype.

They can allow you to COMMIT from inside a SELECT statement. (*)

Incorrect. Refer to Section 8.

Page 5 of 5

You can create a web site application written entirely in PL/SQL. True or False? Mark for Review
(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.

PLSQL feedback final exam semester 1

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.

PLSQL feedback final exam semester 1

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 (*)

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)

Unnamed (*)

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)

PLSQL feedback final exam semester 1

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

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 (*)

In a PL/SQL block, which of the following should not be followed by a semicolon?

Mark for Review

(1) Points

DECLARE (*)

END

PLSQL feedback final exam semester 1

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

PLSQL feedback final exam semester 1

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;
```

PLSQL feedback final exam semester 1

(*)

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

PLSQL feedback final exam semester 1
(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;
```


PLSQL feedback final exam semister 1

```
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 displays 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?

PLSQL feedback final exam semester 1

```
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;
```

PLSQL feedback final exam semester 1

```
SELECT MONTHS_BETWEEN(SYSDATE, '19-MAR-79') /12
AS YEARS
FROM DUAL;
```

None of the above

Which function compares two expressions?
(1) Points

Mark for Review

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

PLSQL feedback final exam semester 1

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;

PLSQL feedback final exam semester 1

```
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);

PLSQL feedback final exam semester 1
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
Page 63

(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 following are scalar data types? (Choose three.)

Mark for

Review

PLSQL feedback final exam semester 1

(1) Points

(Choose all correct answers)

Array

Character (*)

Table

Date (*)

Boolean (*)

(1) Points which of the following is a composite data type?

Mark for Review

CLOB

VARCHAR2

RECORD (*)

DATE

Review (1) Points which of the following are PL/SQL data types? (Choose three.) Mark for Review

(Choose all correct answers)

Large Objects (LOB) (*)

Lexical

Scalar (*)

Delimiter

Composite (*)

(1) Points datatype specifies and restricts the possible data values that can be assigned to a variable. True or False? Mark for Review

True (*)

PLSQL feedback final exam semester 1

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 (*)

PLSQL feedback final exam semester 1
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?
Page 67

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.

PLSQL feedback final exam semester 1

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 (*)

=

<>

PLSQL feedback final exam semester 1

>

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.

PLSQL feedback final exam semester 1

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

PLSQL feedback final exam semester 1
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

Mark for Review

(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

PLSQL feedback final exam semester 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

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 (*)

PLSQL feedback final exam semester 1

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?
(1) Points

Mark for Review

v_family_name = SMITH;

V_FAMILY_NAME = SMITH;

v_family_name := SMITH;

v_family_name := 'SMITH'; (*)

Using implicit conversions is good programming practice.
Review
(1) Points

Mark for

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?
Review
(1) Points

Mark for

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.)
for Review

Mark

(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;

PLSQL feedback final exam semester 1

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
(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.

PLSQL feedback final exam semester 1

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
```

PLSQL feedback final exam semester 1

```
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;

A. 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;
```

END;
Mark for Review
(1) Points

At Line A, code:
v_myvar := 25;

Label both blocks and at line A, code:
v_myvar := 25;

A. It cannot be done because the outer block's v_myvar is out of scope at Line A.

PLSQL feedback final exam semister 1

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 displayedExamine 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

PLSQL feedback final exam semester 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

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

PLSQL feedback final exam semester 1

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. (*)

PLSQL feedback final exam semester 1

The statement will fail because it contains a syntax error.

The row with EMPLOYEE_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

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.

PLSQL feedback final exam semester 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 (*)

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 (*)

PLSQL feedback final exam semester 1

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.

PLSQL feedback final exam semester 1
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?
for Review
(1) Points

Mark

cool cats

big birds and cool cats

aardvarks and cool cats (*)

aardvarks, big birds and cool cats

How many INSERTs can you have in one transaction?
(1) Points

Mark for Review

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.

PLSQL feedback final exam semester 1

Review
(1) Points

In a PL/SQL block, where can you code a COMMIT statement? Mark for

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='Deutschland'
    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 (*)

PLSQL feedback final exam semester 1

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

PLSQL feedback final exam semester 1

which one of the following is correct syntax for an IF statement?
Review

Mark for

(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

PLSQL feedback final exam semester 1

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

"NOT". An error will occur because you cannot combine two Boolean variables using

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';

PLSQL feedback final exam semester 1

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.

PLSQL feedback final exam semester 1

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 (*)

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. (*)

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

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. (*)

PLSQL feedback final exam semester 1
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

PLSQL feedback final exam semister 1

XXXXXXXXXX

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

PLSQL feedback final exam semester 1

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

PLSQL feedback final exam semester 1

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; (*)

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

PLSQL feedback final exam semester 1
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;
```


PLSQL feedback final exam semester 1

```
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 (*)

PLSQL feedback final exam semister 1

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
```

PLSQL feedback final exam semester 1
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

PLSQL feedback final exam semester 1

```
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

PLSQL feedback final exam semester 1

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

PLSQL feedback final exam semester 1

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).

PLSQL feedback final exam semister 1
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) (*)

PLSQL feedback final exam semester 1
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

PLSQL feedback final exam semester 1

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

PLSQL feedback final exam semester 1

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.

PLSQL feedback final exam semister 1

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

PLSQL feedback final exam semester 1
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

PLSQL feedback final exam semester 1

(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);

PLSQL feedback final exam semester 1
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).

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.

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';

PLSQL feedback final exam semister 1
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

PLSQL feedback final exam semester 1

xxxxxxxxxxxxx

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.

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.

PLSQL feedback final exam semester 1

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.

Review 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.

PLSQL feedback final exam semester 1

Create custom reports.

All of the above (*)

Incorrect

Incorrect. Refer to Section 1.

1. Which of the following can be done using PL/SQL?
(1) Points

Mark for Review

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?
(1) Points

Mark for Review

True (*)

False

PLSQL feedback final exam semister 1

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
(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'); (*)

PLSQL feedback final exam semester 1

```
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.)
for Review
(1) Points

Mark

(Choose all correct answers)

Unnamed (*)

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?
for Review
(1) Points

Mark

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.)
for Review
(1) Points

Mark

(Choose all correct answers)

DECLARE

BEGIN (*)

PLSQL feedback final exam semester 1

EXCEPTION

END; (*)

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. 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. 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 (*)

PLSQL feedback final exam semester 1

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

PLSQL feedback final exam semester 1

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. 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. 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. Refer to Section 1.
 Page 134

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 displays 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

PLSQL feedback final exam semister 1
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;
```

(*)

PLSQL feedback final exam semester 1

```
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. Refer to Section 1.
7. which of the following is not a number function?
(1) Points

Mark for Review

TO_DATE (*)

ROUND

MOD

TRUNC

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

PLSQL feedback final exam semester 1

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? Mark for Review
SELECT CONCAT('Today is', 'Thursday!') FROM DUAL;
(1) Points

TodayisThursday!

Today isThursday! (*)

today is thursday!

Today is Thursday!

Incorrect Incorrect. Refer to Section 1.
Page 141

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.
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

PLSQL feedback final exam semester 1

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

Review

(1) Points

PLSQL feedback final exam semester 1

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.)
Review
(1) Points

Mark for

(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?
Review
(1) Points

Mark for

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 following are scalar data types? (Choose three.) Mark for Review
(1) Points

(Choose all correct answers)

Array

Character (*)

Table

Date (*)

Boolean (*)

Incorrect. Refer to Section 2.
5. which of the following are scalar data types? (Choose three.) Mark for Review
(1) Points

(Choose all correct answers)

Array

Character (*)

Table

Date (*)

Boolean (*)

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 (*)

PLSQL feedback final exam semester 1

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

PLSQL feedback final exam semester 1

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. 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. 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.

PLSQL feedback final exam semester 1

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

PLSQL feedback final exam semester 1

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;
```

PLSQL feedback final exam semester 1

(*)

```
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;

PLSQL feedback final exam semester 1
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

PLSQL feedback final exam semester 1

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 (*)

PLSQL feedback final exam semester 1

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

PLSQL feedback final exam semester 1

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
Page 159

PLSQL feedback final exam semister 1

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');

PLSQL feedback final exam semester 1
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
Page 161

(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;

PLSQL feedback final exam semister 1

A. It cannot be done because the outer block's v_myvar is out of scope at Line

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.

PLSQL feedback final exam semester 1
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. 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. 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. Refer to Section 3.
2. To modify an existing row in a table, you can use the _____ statement.
Mark for Review

(1) Points

PLSQL feedback final exam semester 1

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

PLSQL feedback final exam semester 1
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 EMPLOYEE_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
Page 168

PLSQL feedback final exam semester 1
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

PLSQL feedback final exam semister 1

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

(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.

PLSQL feedback final exam semester 1

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".

PLSQL feedback final exam semester 1

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

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. (*)

PLSQL feedback final exam semester 1
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.

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='Deutschland'
    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

PLSQL feedback final exam semester 1

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

PLSQL feedback final exam semester 1

(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. 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. 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. 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

PLSQL feedback final exam semester 1

False (*)

NULL

Undefined

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. 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

PLSQL feedback final exam semester 1

Equal

Undefined (*)

Unequal

Nothing will be displayed because V_MESSAGE is set to NULL.

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. 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

PLSQL feedback final exam semester 1

WHEN 90 THEN v_grade := 'A'

WHEN 90 THEN v_grade := 'A'; (*)

WHEN 90 THEN 'A'

WHEN 90 THEN 'A';

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. 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 (*)

PLSQL feedback final exam semester 1
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. (*)

PLSQL feedback final exam semester 1
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. 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. 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?
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.
Page 184

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

PLSQL feedback final exam semester 1

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. 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.
Page 188

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. 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)

PLSQL feedback final exam semester 1

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

PLSQL feedback final exam semester 1

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

PLSQL feedback final exam semester 1

```
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. 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));
```

PLSQL feedback final exam semister 1

```
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:

PLSQL feedback final exam semester 1

```
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 ...

PLSQL feedback final exam semester 1
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 ...

PLSQL feedback final exam semester 1

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. 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. 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 ...

PLSQL feedback final exam semester 1

```
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
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;
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
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
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. 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. 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

PLSQL feedback final exam semester 1

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)`

PLSQL feedback final exam semester 1

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.

PLSQL feedback final exam semester 1

You cannot OPEN more than one cursor at the same time.

None of the above, they are all allowed. (*)

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. 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. 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
```

PLSQL feedback final exam semester 1

```
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;

PLSQL feedback final exam semester 1

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.
variables defined?
(1) Points

In which part of the PL/SQL block are declarations of
Mark for Review

Executable

Exception

Declarative (*)

Definition

Incorrect

Incorrect. Refer to Section 1.

3.
test PL/SQL code?
(1) Points

which of the following tools can NOT be used to develop and
Mark for Review

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?
(1) Points Mark for Review

Application Builder

SQL workshop (*)

Utilities

Object Browser PLSQL feedback final exam semester 1

Incorrect Incorrect. Refer to Section 1.

Review 5. which PL/SQL block type must return a value? Mark for
(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.

PL/SQL block? 7. what is the purpose of using DBMS_OUTPUT.PUT_LINE in a
(1) Points Mark for Review

To perform conditional tests

To allow a set of statements to be executed repeatedly

PLSQL feedback final exam semester 1

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

To store new rows in the database

Incorrect

Incorrect. Refer to Section 1.

for Review
(1) Points

8. which of the following can you use PL/SQL to do?

Mark

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?
(1) Points

Mark for Review

True

False (*)

Correct

Correct

Mark for Review
(1) Points

10. The fact that PL/SQL is portable is a good thing because:

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 (*)

PLSQL feedback final exam semester 1

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

PLSQL feedback final exam semester 1

procedural (*)

low level

Incorrect

Incorrect. Refer to Section 1.

Section 2

or False?
(1) Points

14. A variable must have a value if NOT NULL is specified. True
Mark for Review

True (*)

False

Incorrect

Incorrect. Refer to Section 2.

for Review
(1) Points

15. which of the following declarations is invalid? Mark

v_count PLS_INTEGER:=0;

college_name VARCHAR2(20):='Harvard';

v_pages CONSTANT NUMBER; (*)

v_start_date DATE := sysdate+1;

Correct

Correct

variable?
(1) Points

16. which of the following should NOT be used as the name of a
Mark for Review

A table name.

A table column name. (*)

The database name.

PLSQL feedback final exam semester 1

Correct

Correct

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

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?
(1) Points Mark for Review

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; (*)

PLSQL feedback final exam semester 1

```
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.

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

PLSQL feedback final exam semester 1

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 PLSQL feedback final exam semister 1
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

PLSQL feedback final exam semester 1

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)

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

PLSQL feedback final exam semester 1
students_street_address (*)

v_code (*)

#hours

completion_%

Incorrect

Incorrect. Refer to Section 2.

for Review
(1) Points

33. which statement most closely describes "data type"? Mark

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

for Review
(1) Points

34. _____ are meant to store large amounts of data. Mark

Variables

Scalar data types

LOBs (*)

Incorrect

Incorrect. Refer to Section 2.

for Review
(1) Points

35. A movie is an example of which category of data type? Mark

Scalar

PLSQL feedback final exam semester 1

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

PLSQL feedback final exam semester 1
(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.

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

PLSQL feedback final exam semester 1

```
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

PLSQL feedback final exam semester 1

True

False (*)

Correct

Correct

for Review
(1) Points

49. which SQL statement can NOT use an implicit cursor?

Mark

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

PLSQL feedback final exam semester 1

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

(1) Points

5. The P in PL/SQL stands for: Mark for Review

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. PLSQL feedback final exam semister 1
variables defined? In which part of the PL/SQL block are declarations of
(1) Points Mark for Review

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'); (*)

PLSQL feedback final exam semester 1

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. (*)

PLSQL feedback final exam semester 1
The database name.

Correct

Correct

database.
(1) Points

17. Delimiters are _____ that have special meaning to the Oracle
Mark for Review

identifiers

variables

symbols (*)

Correct

Correct

Mark for Review
(1) Points

18. which of the following are valid identifiers? (Choose two.)

(Choose all correct answers)

Full Name

students_street_address (*)

v_code (*)

#hours

completion_%

Correct

Correct

Mark for Review
(1) Points

19. which statements about lexical units are true? (Choose two.)

(Choose all correct answers)

They are named objects stored in the database

They are the building blocks of every PL/SQL program (*)

PLSQL feedback final exam semister 1

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 (*)

PLSQL feedback final exam semester 1

LOB

Incorrect

Incorrect. Refer to Section 2.

for Review
(1) Points

27. _____ are meant to store large amounts of data.

Mark

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

PLSQL feedback final exam semester 1

(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

PLSQL feedback final exam semester 1

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;
```

PLSQL feedback final exam semister 1

```
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.

PLSQL feedback final exam semester 1

Correct

Correct

transaction? 46. How many DML statements can be included in a single
(1) Points Mark for Review

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 (*)

PLSQL feedback final exam semester 1
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

PLSQL feedback final exam semester 1

Basic loop (*)

WHILE loop

CASE loop

Correct

Correct

statement? 8. Which one of these tasks is best done using a LOOP
(1) Points Mark for Review

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

Review
(1) Points

10. which one of these is NOT a kind of loop? Mark for

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

```

PLSQL feedback final exam semester 1

```
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 should be used? Mark for Review
(1) Points

PLSQL feedback final exam semester 1

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

PLSQL feedback final exam semester 1

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;  
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.
Page 253

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.

PLSQL feedback final exam semester 1

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

Correct

Correct

you use?
(1) Points

27. When using multiple nested cursors, what kinds of loops can
Mark for Review

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;

PLSQL feedback final exam semester 1
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;
  ...
```


PLSQL feedback final exam semester 1

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 ...);
  &nbsp;...
```

To display the fetched last name, what should you code at Point A?

Mark for Review

(1) Points

v_emp_rec.last_name (*)

PLSQL feedback final exam semister 1

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
    emp_curs CURSOR IS
        SELECT job_id FROM employees;
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.

PLSQL feedback final exam semester 1

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;
```

PLSQL feedback final exam semester 1

```
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;
```

PLSQL feedback final exam semister 1

```
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.

PLSQL feedback final exam semester 1

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

PLSQL feedback final exam semester 1
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.

cursor?
(1) Points

49. What is one of the advantages of using parameters with a cursor?
Mark for Review

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 (*)

PLSQL feedback final exam semester 1

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

PLSQL feedback final exam semister 1
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

PLSQL feedback final exam semester 1

anonymous (*)

Incorrect. Refer to Section 1.

(Choose two.) 8. Which keywords must be included in every PL/SQL block?
(1) Points Mark for Review

(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)

PLSQL feedback final exam semester 1

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

PLSQL feedback final exam semester 1

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

PLSQL feedback final exam semester 1

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

PLSQL feedback final exam semester 1

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.

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

PLSQL feedback final exam semester 1

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)

PLSQL feedback final exam semester 1

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

for Review
(1) Points

34. _____ are meant to store large amounts of data. Mark

variables

scalar data types

LOBs (*)

Correct

for Review
(1) Points

35. A collection is a composite data type. True or False? Mark

True (*)

False

Correct

labeled?
(1) Points

36. When nested blocks are used, which blocks can or must be labeled? Mark for Review

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

PLSQL feedback final exam semester 1

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.

PLSQL feedback final exam semester 1

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.

PLSQL feedback final exam semester 1

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_ems (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.

PLSQL feedback final exam semister 1

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. 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

2. PL/SQL is an Oracle proprietary, procedural, 4GL programming language.
True or False?

Mark for Review

(1) Points

PLSQL feedback final exam semester 1

True

False (*)

Incorrect. Refer to Section 1

3.
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.

Section 2

4.
Assignment statements can continue over several lines in PL/SQL. True or False?
Mark for Review
(1) Points

True (*)

False

Incorrect. Refer to Section 2.

5.
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

Section 2

6.
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

PLSQL feedback final exam semister 1

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. Refer to Section 2.

7.
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. Refer to Section 2.

8. _____ are meant to store large amounts of data.
(1) Points

Mark for Review

Variables

Scalar data types

LOBs (*)

Incorrect. Refer to Section 2.

9.
Which of the following are valid identifiers? (Choose two.)
(1) Points

Mark for Review

(Choose all correct answers)

yesterday (*)

yesterday's date

number_of_students_in_the_class

v\$testresult (*)

#students

Incorrect. Refer to Section 2.

Section 3

10.

You declare an implicit cursor in the DECLARE section of a PL/SQL block. True or False? Mark for Review

(1) Points

True

False (*)

Section 3

11.

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. Refer to Section 3.

12.
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

Incorrect. Refer to Section 3.

Section 4

13.
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.

14.

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.

15.
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.

Section 4

16.

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.

17.

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

6

30 (*)

40

Incorrect. Refer to Section 4.

Section 5

18.
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.

19.

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.

20.

Which of the following cursor attributes evaluates to TRUE if the cursor is open?

Mark for Review

(1) Points

%ISOPEN (*)

%NOTFOUND

%FOUND

%ROWCOUNT

Section 5

21.

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

22.

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;

PLSQL feedback final exam semester 1

DELETE FROM departments WHERE c%ROWCOUNT = 1;

None of the above.

Incorrect. Refer to Section 5.

23.
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.

Section 6

24.
Examine the following code. What is the scope and visibility of the outer block's v_last_name?
DECLARE
 v_last_name VARCHAR2(20);
BEGIN
 DECLARE
 v_last_name VARCHAR2(20);
 BEGIN
 ...
 END;
 ...
END; ... Mark for Review
(1) Points

It is in scope and visible in both blocks.

It is in scope and visible in the outer block only.

It is in scope in both blocks, but visible only in the outer block. (*)

It is visible in both blocks, but in scope only in the outer block.

Incorrect. Refer to Section 6.

25.

There are no employees in department 99. what message or messages will be displayed when the following code is executed?

```
DECLARE
    e_my_excep EXCEPTION;
BEGIN
    BEGIN
        UPDATE employees SET salary = 10000
            WHERE department_id = 99;
        IF SQL%ROWCOUNT = 0 THEN
            RAISE e_my_excep;
        END IF;
    EXCEPTION
        WHEN e_my_excep THEN
```

```

                PLSQL feedback final exam semister 1
        DBMS_OUTPUT.PUT_LINE('Message 1');
        RAISE e_my_excep;
        DBMS_OUTPUT.PUT_LINE('Message 2');
    END;
    DBMS_OUTPUT.PUT_LINE('Message 3');
EXCEPTION
    WHEN e_my_excep THEN
        DBMS_OUTPUT.PUT_LINE('Message 4');
END;
Mark for Review
(1) Points

```

Message 1
Message 3

Message 1
Message 2

Message 1
Message 3
Message 4

Message 1
Message 4
(*)

Section 6

26.
Which of the following are good practice guidelines for exception handling? (Choose three.)
Mark for Review
(1) Points

PLSQL feedback final exam semester 1

(Choose all correct answers)

Test your code with different combinations of data to see what potential errors can happen. (*)

Use an exception handler whenever there is any possibility of an error occurring. (*)

Include a WHEN OTHERS handler as the first handler in the exception section.

Allow exceptions to propagate back to the calling environment.

Handle specific named exceptions where possible, instead of relying on WHEN OTHERS. (*)

Incorrect. Refer to Section 6.

27.
which of the following are examples of predefined Oracle server errors? (Choose three.)
(1) Points

Mark for Review

(Choose all correct answers)

TOO_MANY_ROWS (*)

NO_DATA_FOUND (*)

OTHERS

ZERO_DIVIDE (*)

E_INSERT_EXCEP

Incorrect. Refer to Section 6.

Section 7

28.

The following procedure has been created:

```
CREATE OR REPLACE PROCEDURE defproc
```

```
(A IN NUMBER := 50,
```

```
B IN NUMBER,
```

```
C IN NUMBER DEFAULT 40)
```

```
IS .....
```

Which one of the following will invoke the procedure correctly?

Mark for

Review

(1) Points

```
defproc(30 => A);
```

```
defproc(30, 60 => C);
```

```
defproc(40, 70); (*)
```

PLSQL feedback final exam semester 1

```
defproc(10 => A, 25 => C);
```

```
defproc;
```

Incorrect. Refer to Section 7.

29.
which of the following can NOT be used as the datatype of a procedure parameter?
Mark for Review
(1) Points

A non-SQL datatype such as BOOLEAN

The name of another procedure (*)

A large object datatype such as CLOB

A PLSQL record defined using %ROWTYPE

Incorrect. Refer to Section 7.

30.
The following are the steps involved in creating, and later modifying and re-creating, a PL/SQL procedure in Application Express. In what sequence should these steps be performed?
- A. Retrieve the saved code from "Saved SQL" in SQL Commands
 - B. Execute the code to create the procedure
 - C. Execute the code to re-create the procedure
 - D. Click on the "Save" button and save the procedure code
 - E. Modify the code in the SQL Commands window
 - F. Type the procedure code in the SQL Commands window
- (1) Points Mark for Review

F,C,A,B,E,D

F,B,D,A,E,C (*)

E,D,F,C,A,B

PLSQL feedback final exam semister 1

F,B,D,E,A,C

F,B,C,D,E,A

Section 8

31.
How do you specify that you want a procedure MYPROCA to use Invoker's Rights?
Mark for Review
(1) Points

```
CREATE OR REPLACE PROCEDURE myproca  
AUTHID CURRENT_USER IS...  
(*)
```

Invoker's Rights are the default, therefore no extra code is needed.

```
GRANT INVOKER TO myproca;
```

```
ALTER PROCEDURE myproca TO INVOKER;
```

```
CREATE OR REPLACE PROCEDURE myproca  
AUTHID OWNER IS...
```


Incorrect. Refer to Section 8.

32.
In which DML statements can user-defined functions be used? Mark for Review
(1) Points

INSERT and UPDATE, but not DELETE.

INSERT only.

All DML statements. (*)

UPDATE only

DELETE only

Incorrect. Refer to Section 8.

33.
Which of the following is a difference between a procedure and a function?
Mark for Review
(1) Points

Functions cannot be nested; procedures can be nested to at least 8 levels.

A procedure can have default values for parameters, while a function cannot.

An explicit cursor can be declared in a procedure, but not in a function.

A function cannot be used within a SQL statement; a procedure can be used within SQL.

A function must return a value, a procedure may or may not. (*)

Incorrect. Refer to Section 8.

34.

You want to remove the procedure NO_NEED from your schema. You execute:
`DROP PROCEDURE no_need;`
which Data Dictionary views are updated automatically? Mark for Review
(1) Points

USER_PROCEEDURES

USER_OBJECTS

USER_SOURCE

All of the above. (*)

None of the above.

Incorrect. Refer to Section 8.

Section 9

35.

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

Section 10

41.
What is wrong with the following code?

```
CREATE OR REPLACE TRIGGER loc_trigg
BEFORE DELETE ON locations
BEGIN
    RAISE_APPLICATION_ERROR(-20201,'Invalid delete');
    ROLLBACK;
END;
```

Mark for Review
(1) Points

The last line should be:
END loc_trigg;

You cannot use RAISE_APPLICATION_ERROR inside a trigger.

The second line should be:
BEFORE DELETE OF locations

You cannot use ROLLBACK inside a trigger.
(*)

PLSQL feedback final exam semester 1

Nothing is wrong, this trigger will compile and execute successfully.

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?
(1) Points

Mark for Review

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; (*)

PLSQL feedback final exam semester 1

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

None. All of the above are allowed.

Incorrect. Refer to Section 10.

43.
Which dictionary view shows the detailed code of a trigger body?
Review
(1) Points

Mark for

USER_SOURCE

USER_TRIGGERS (*)

USER_OBJECTS

PLSQL feedback final exam semester 1

USER_DML_TRIGGERS

USER_SUBPROGRAMS

Incorrect. Refer to Section 10.

44.

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.

45.
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

Section 11

46.

A PL/SQL package named MYPACK declares a record type named MYTYPE as a public variable in the package specification. Which of the following anonymous blocks successfully declares a local variable of datatype MYTYPE? Mark for Review
(1) Points

```
DECLARE
    v_myrec IS RECORD mypack.mytype;
BEGIN ...
```

```
DECLARE
    v_myrec mypack.mytype;
BEGIN ...
(*)
```

```
DECLARE
    v_myrec mytype;
BEGIN ...
```

PLSQL feedback final exam semester 1

```
DECLARE
    v_myrec IS RECORD (mypack.mytype);
BEGIN ...
```

Incorrect. Refer to Section 11.

47.

Examine the following code:

```
DECLARE
    CURSOR emp_curs IS
        SELECT employee_id, first_name, last_name FROM employees;
    TYPE t_mytype IS TABLE OF -- Point A
        INDEX BY BINARY_INTEGER;
    v_mytab t_mytype;
```

Which of the following can be coded at Point A?

Mark for Review

(1) Points

employees%ROWTYPE

employees.salary%TYPE

emp_curs%ROWTYPE

Any one of the above (*)

None of the above

Incorrect. Refer to Section 11.

48.

The database administrator has created a directory as follows:

```
CREATE DIRECTORY filesdir AS 'C:\BFILEDIR';
```

How would the DBA allow all database users to query the BFILES in this directory?

Mark for Review

(1) Points

```
GRANT READ ON filesdir TO PUBLIC;
```

```
GRANT READ ON DIRECTORY filesdir TO PUBLIC; (*)
```

```
GRANT SELECT ON filesdir TO PUBLIC;
```

PLSQL feedback final exam semester 1

```
GRANT QUERY ON DIRECTORY filesdir TO PUBLIC;
```

```
GRANT READ ON 'C:\BFILEDIR' TO PUBLIC;
```

Incorrect. Refer to Section 11.

49.

Which of the following methods can be used to query CLOB data values? (Choose two.)

Mark for Review

(1) Points

(Choose all correct answers)

```
SELECT (*)
```

PLSQL feedback final exam semister 1

DBMS_LOB.PUT

DBMS_LOB.GETLENGTH

DBMS_LOB.READ (*)

Incorrect. Refer to Section 11.

Section 12

50.

Examine the following code:

```
CREATE FUNCTION deptfunc  
RETURN NUMBER IS  
    v_count NUMBER(6);  
BEGIN  
    SELECT COUNT(*) INTO v_count FROM departments;  
    RETURN v_count;  
END;
```

Which of the following will display the dependency between DEPTFUNC and DEPARTMENTS?

Mark for Review

(1) Points

```
SELECT name, type
FROM user_dependencies
WHERE name IN ('DEPTFUNC', 'DEPARTMENTS');
```

```
SELECT name, type, referenced_name, referenced_type
FROM user_dependencies
WHERE referenced_name = 'DEPARTMENTS'
AND referenced_type = 'TABLE';
(*)
```

```
SELECT name, type, referenced_name, referenced_type
FROM user_dependencies
WHERE name = 'DEPARTMENTS'
AND type = 'TABLE';
```

```
SELECT object_name, object_type
FROM user_objects
WHERE object_name IN ('DEPARTMENTS', 'DEPTFUNC')
AND referenced = 'YES';
```