Any variable declared in the Stored Package Specification may be accessed from:

a.Anywhere outside or inside that Package

b. Only from inside the Package

c. Only from SQL\*PLUS

d. Just from Stored Package Procedures

If certain boolean function was NOT declared in a Specification of the Package, then what statement is correct?

a. This function may be called outside that package only through an assignment to a global variable

b. This function must be called through a SELECT or DML statement

c. Function is private and can not be called outside the package

d. Function is global and any statement may call it

Evaluate this cursor: Cursor c1 (p\_salary Number(8,2)) IS Select employee\_id, job\_id, hire\_date From employees Where salary > v\_salary; Why does this statement cause an error:

a.A parameter is missing IN, OUT or IN OUT specification  
b.The Select statement is missing the INTO clause

c.The WHERE clause can’t be used in a CURSOR statement

d.The size of a parameter can not be specified

If you want to see how Procedure update\_emp was created, the right query in SQL\*PLUS is:  
a.SELECT text FROM user\_objects WHERE object\_name = ‘UPDATE\_EMP’; b.SELECT line FROM user\_source WHERE name = ‘update\_emp’;

c.SELECT line FROM user\_procedures WHERE name = ‘UPDATE\_EMP’;

d.SELECT text FROM user\_source WHERE name = ‘UPDATE\_EMP’;

CREATE OR REPLACE PACKAGE TESTING -- Line1 (p\_var1 IN varchar2) -- Line2 IS -- Line3 :global\_var : = 10; -- Line4 PROCEDURE MYTEST ( -- Line5  
p\_sal EMPLOYEES.SALARY%TYPE) IS -- Line6 Etc.  
There are some errors in the Package Spec code above? What are their line numbers?

a.2,3,4,6

b.2,4,6

c.2,6

d.4, 6

CREATE FUNCTION get\_emp\_job (

p\_empid NUMBER) RETURN VARCHAR2 IS

v\_job VARCHAR2(15);

BEGIN ...

RETURN v\_job; ...

END;

Below are shown four execution examples for a Function declared above, by using various Methods. Which Method is not correct one.?

a.

EXECUTE dbms\_output.put\_line.get\_emp\_job(201);

-----------------------------------------------------------------------

b.

VARIABLE job VARCHAR2(15)

EXECUTE :job := get\_emp\_job(201)

PRINT job

--------------------------------------------------

c.

DECLARE

v\_job VARCHAR2(15);

BEGIN

v\_job:= get\_emp\_job (201);

DBMS\_OUTPUT.PUT\_LINE(v\_job);

END;

-----------------------------------------------------------

d.

SELECT get\_emp\_job(201) FROM dual;

-------------------------------------------------------

Which statement about Procedures is not correct one:

a. Procedures can take IN , OUT and IN OUT parameters

b. Procedures are usually used to perform an action

c. Procedures may return One or More values, and may not return zero values

d. You may execute Procedure stand alone, by just calling its name in the Block

CREATE OR REPLACE PROCEDURE mine( p\_In IN NUMBER, p\_Out OUT NUMBER, p\_InOut IN OUT NUMBER) IS v\_local NUMBER := 10; BEGIN Which one of the following statements will not work in the executable section of the following procedure?

1. v\_local := v\_local + p\_In;
2. p\_In := v\_local;
3. p\_InOut := v\_local;
4. p\_Out := p\_In + v\_local;

Which statement about Functions is not correct one?

a. Function must have RETURN Datatype in the Header before IS

b. Functions must have exactly one "RETURN value" statement

c. Functions can have "RETURN value" in the EXCEPTION section

d. Functions always return a single value

CREATE PROCEDURE modify\_sales\_rep ( p\_empid NUMBER, p\_sal NUMBER default 5000, p\_com NUMBER default 0.2) IS ... Below are shown four execution examples for a Procedure declared above, by using various Notations with Actual parameters. Which example is not correct one.?

1. EXECUTE modify\_sales\_rep (301);
2. EXECUTE modify\_sales\_rep (p\_sal=>6000, p\_empid=>301);
3. EXECUTE modify\_sales\_rep (p\_sal=>6000, p\_com=>0, p\_empid=>301);
4. EXECUTE modify\_sales\_rep (p\_sal=>6000, 301);

You want to use Forward Declaration in your Package Body called my\_pack with Two Procedures called m\_proc and t\_proc. Both procedures have no arguments. Procedure t\_proc is called by procedure m\_proc. What is the correct flow of this Package Body (with no code in the executable section), so that your two Procedure Bodies are listed in the alphabetical order.

a.

PROCEDURE m\_proc;

PROCEDURE m\_proc IS ... END;

PROCEDURE t\_proc IS ... END;

-----------------------------------------------

b.

PROCEDURE t\_proc;

PROCEDURE m\_proc IS ... END;

PROCEDURE t\_proc IS ... END;

-------------------------------------------------

c.

PROCEDURE m\_proc;

PROCEDURE t\_proc IS ... END;

PROCEDURE m\_proc IS ... END;

------------------------------------------------

d.

PROCEDURE t\_proc;

PROCEDURE t\_proc IS ... END;

PROCEDURE m\_proc IS ... END;

What type of DML trigger regarding Timing does not exist?

a. Before

b. After

c. On

d. Instead Of