**\\Practice 2**

* + 1. Which of the following PL/SQL blocks execute successfully?

a. BEGIN

END; +

b. DECLARE

amount INTEGER(10);

END;

c. DECLARE

BEGIN

END;

d. DECLARE

amount INTEGER(10);

BEGIN

DBMS\_OUTPUT.PUT\_LINE(amount);

END;

1. Create and execute a simple anonymous block that outputs “Hello World.”

BEGIN

DBMS\_OUTPUT.PUT\_LINE('Hello World');

END;

3. Evaluate each of the following declarations. Determine which of them are not legal and explain why.

a. DECLARE

name,dept VARCHAR2(14);

b. DECLARE

test NUMBER(5);

c. DECLARE

MAXSALARY NUMBER(7,2) = 5000;

d. DECLARE

JOINDATE BOOLEAN := SYSDATE; +

4. In each of the following assignments, determine the data type of the resulting expression.

a. email := firstname || to\_char(empno); string

b. confirm := to\_date('20-JAN-1999', 'DD-MON-YYYY'); date

c. sal := (1000\*12) + 500 numbers

d. test := FALSE; Boolean

e. temp := temp1 < (temp2/ 3); Boolean

f. var := sysdate; date

5. Identify valid and invalid identifier names:

a. today

b. last\_name

c. today’s\_date X

d. Number\_of\_days\_in\_February\_this\_year

e. Isleap$year

f. #number X

g. NUMBER# X

h. number1to7

6. Identify valid and invalid variable declaration and initialization:

a. number\_of\_copies PLS\_INTEGER;

b. printer\_name constant VARCHAR2(10);

c. deliver\_to VARCHAR2(10):=Johnson; X

d. by\_when DATE:= SYSDATE+1;

7. Examine the following anonymous block and choose the appropriate statement.

SET SERVEROUTPUT ON

DECLARE

fname VARCHAR2(20);

lname VARCHAR2(15) DEFAULT 'fernandez';

BEGIN

DBMS\_OUTPUT.PUT\_LINE( FNAME ||' ' ||lname);

END;

/

* + - 1. The block will execute successfully and print ‘fernandez’
      2. The block will give an error because the fname variable is used without initializing.
      3. The block will execute successfully and print ‘null fernandez’ V
      4. The block will give an error because you cannot use the DEFAULT keyword to initialize a variable of type VARCHAR2.
      5. The block will give an error because the variable FNAME is not declared.

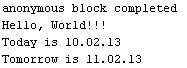
8. Create an anonymous block as in question 2 of practice 1.

* + - 1. Add a declarative section to this PL/SQL block. In the declarative section, declare the following variables:

1. Variable today of type DATE. Initialize today with SYSDATE.

2. Variable tomorrow of type today. Use %TYPE attribute to declare this variable.

1. In the executable section initialize the variable tomorrow with an expression, which calculates tomorrow’s date (add one to the value in today). Print the value of today and tomorrow after printing ‘Hello World’.



DECLARE

today DATE := SYSDATE;

tomorrow today%TYPE;

BEGIN

tomorrow := today + 1;

DBMS\_OUTPUT.PUT\_LINE('Hello, world!');

DBMS\_OUTPUT.PUT\_LINE('Today is ' || TO\_CHAR(today, 'DD.MM.YY'));

DBMS\_OUTPUT.PUT\_LINE('Tomorrow is ' || TO\_CHAR(tomorrow, 'DD.MM.YY'));

END;

/

9. Evaluate the PL/SQL block given below and determine the data type and value of each of the following variables according to the rules of scoping.

a. The value of weight at position 1 is: NUMBER(3) := 600;

b. The value of new\_locn at position 1 is: не обьявлен

c. The value of weight at position 2 is: NUMBER(3) := 1;

d. The value of message at position 2 is: VARCHAR2(255) := 'Product 11001';

e. The value of new\_locn at position 2 is: VARCHAR2(50) := 'Europe';

**PL/SQL Block**

DECLARE

weight NUMBER(3) := 600;

message VARCHAR2(255) := 'Product 10012';

BEGIN

DECLARE

weight NUMBER(3) := 1;

message VARCHAR2(255) := 'Product 11001';

new\_locn VARCHAR2(50) := 'Europe';

BEGIN

weight := weight + 1;

new\_locn := 'Western ' || new\_locn;

END;  
weight := weight + 1;

message := message || ' is in stock';

new\_locn := 'Western ' || new\_locn;

END;

10. In the PL/SQL block shown below, determine the values and data types for each of the following cases.

a. The value of customer in the nested block is: NUMBER(7) := 201;

b. The value of c\_name in the nested block is: VARCHAR2(25) := 'Unisports';

c. The value of credit\_rating in the nested block is: credit\_rating :='GOOD';

d. The value of customer in the main block is: VARCHAR2(50) := 'Womansport';

e. The value of c\_name in the main block is: не объявлено

f. The value of credit\_rating in the main block is: VARCHAR2(50) := 'EXCELLENT';

**Scope Example**

DECLARE

customer VARCHAR2(50) := 'Womansport';

credit\_rating VARCHAR2(50) := 'EXCELLENT';

BEGIN  
 DECLARE  
 customer NUMBER(7) := 201;

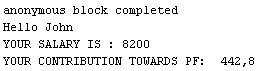
c\_name VARCHAR2(25) := 'Unisports';

BEGIN  
 credit\_rating :='GOOD';

…  
 END;  
 …  
END;

11. You need to create an anonymous block to output the employee’s name, his salary and his contribution towards provident fund (PF).

* + - 1. Declare two variables: fname of type VARCHAR2 and size 15, and emp\_sal of type NUMBER and size.
      2. Include the SQL statement in the executable section that retrieve the name and the salary of the employee with id equals 100 into the variables declared above.
      3. Output the line ‘Hello’ and the first name of the employee.
      4. Calculate the contribution of the employee towards provident fund (PF).  
         PF is 12% of the basic salary and basic salary is 45% of the salary. Try and use only one expression to calculate the PF. Print the employee’s salary and his contribution towards PF.
      5. Execute your script. Sample output is shown below.



DECLARE

fname VARCHAR2(15);

emp\_sal NUMBER;

pf\_contrib NUMBER;

BEGIN

SELECT FIRST\_NAME, SALARY

INTO fname, emp\_sal

FROM hr.employees

WHERE EMPLOYEE\_ID = 100;

DBMS\_OUTPUT.PUT\_LINE('Hello ' || fname);

pf\_contrib := 0.12 \* (0.45 \* emp\_sal);

DBMS\_OUTPUT.PUT\_LINE('YOUR SALARY IS: ' || emp\_sal);

DBMS\_OUTPUT.PUT\_LINE('YOUR CONTRIBUTION TOWARDS PF: ' || pf\_contrib);

END;

/