**Practice 2**

1. Identify valid and invalid identifier names:

a. today

b. last\_name

c. today’s\_date

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

e. Isleap$year

f. #number

g. NUMBER#

h. number1to7

2. 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;

d. by\_when DATE:= CURRENT\_DATE+1;

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

DECLARE

v\_fname VARCHAR2(20);

v\_lname VARCHAR2(15) DEFAULT 'fernandez';

BEGIN

DBMS\_OUTPUT.PUT\_LINE(v\_fname ||' ' ||v\_lname);

END;

/

a. The block executes successfully and print “fernandez.”

b. The block returns an error because the fname variable is used without initializing.

c. The block executes successfully and print “null fernandez.”

d. The block returns an error because you cannot use the DEFAULT keyword to initialize a variable of type VARCHAR2.

e. The block returns an error because the v\_fname variable is not declared.

4. Create an anonymous block. In SQL Developer, load the lab\_01\_02\_soln.sql script, which you created in question 2 of practice 1.

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

1. Variable v\_today of type DATE. Initialize today with SYSDATE.

2. Variable v\_tomorrow of type today. Use %TYPE attribute to declare this variable.

b.In the executable section, initialize the tomorrow variable 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.”

c. Execute and save this script as lab\_02\_04\_soln.sql. Sample output is as follows:

set serveroutput on;

declare

v\_today DATE := sysdate;

v\_tomorrow v\_today%type;

v\_line varchar(15):= 'Hello world';

begin

v\_tomorrow := v\_today + 1;

dbms\_output.put\_line(v\_line);

dbms\_output.put\_line(v\_today ||' '||v\_tomorrow);

end;

Edit the lab\_02\_04\_soln.sql script.

a. Add code to create two bind variables.   
Create bind variables b\_basic\_percent and b\_pf\_percent of type NUMBER.

b. In the executable section of the PL/SQL block, assign the values 45 and 12 to b\_basic\_percent and b\_pf\_percent, respectively.

c. Terminate the PL/SQL block with “/” and display the value of the bind variables by using the PRINT command.

set serveroutput on ;

set autoprint on;

variable b\_basic\_percent number;

variable b\_pf\_percent number;

begin

:b\_basic\_percent:= 45;

:b\_pf\_percent:=12;

end;

/

print b\_basic\_percent;

print b\_pf\_percent;

* + - 1. Execute and save your script file as lab\_02\_05\_soln.sql. Sample output is as follows:

