ITE 5220 Oracle Database Programming using PL/SQL

Lab Exercise 2[Chapter 3]

6 POINTS

Agenda:

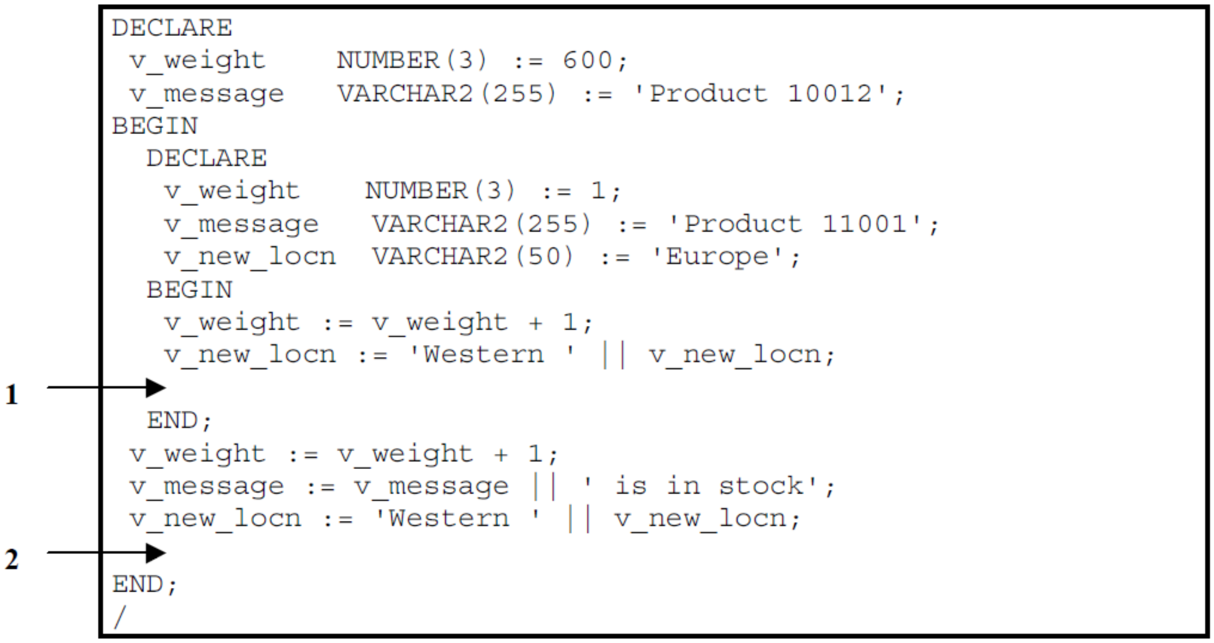
To do this lab you will have to use the computers in the lab.

You have to capture the output and write your findings about the output.

Practice 1: Writing Executable Statements[2 Points]

In this practice, you examine and write executable statements.

a)



```
DECLARE
 v_weight    NUMBER(3) := 600;
 v_message   VARCHAR2(255) := 'Product 10012';
BEGIN
  DECLARE
   v_weight    NUMBER(3) := 1;
   v_message   VARCHAR2(255) := 'Product 11001';
   v_new_locn  VARCHAR2(50) := 'Europe';
  BEGIN
   v_weight := v_weight + 1;
   v_new_locn := 'Western ' || v_new_locn;

  END;
 v_weight := v_weight + 1;
 v_message := v_message || ' is in stock';
 v_new_locn := 'Western ' || v_new_locn;

END;
/
```

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

a) The value of v_weight at position 1 is:

b) The value of v_new_locn at position 1 is:

c) The value of v_weight at position 2 is:

d) The value of v_message at position 2 is:

e) The value of v_new_locn at position 2 is:

b)

```
DECLARE
   v_customer     VARCHAR2(50) := 'Womansport';
   v_credit_rating   VARCHAR2(50) := 'EXCELLENT';
BEGIN
    DECLARE
       v_customer  NUMBER(7) := 201;
       v_name VARCHAR2(25) := 'Unisports';
    BEGIN
       v_credit_rating :='GOOD';
       …
    END;
  …
END;
```

In the preceding PL/SQL block, determine the values and data types for each of the following cases:

a) The value of v_customer in the nested block is:

b) The value of v_name in the nested block is:

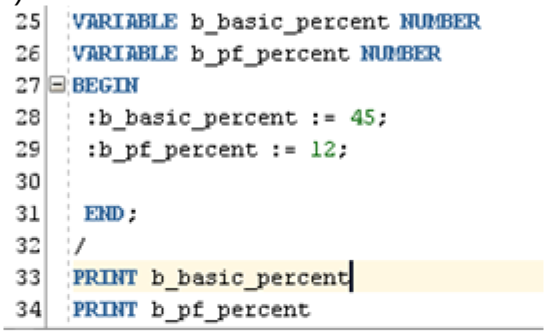
c) The value of v_credit_rating in the nested block is:

d) The value of v_customer in the main block is:

e) The value of v_name in the main block is:

f) The value of v_credit_rating in the main block is:

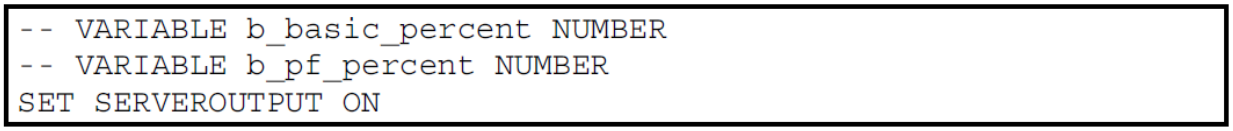
c)



```
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
```

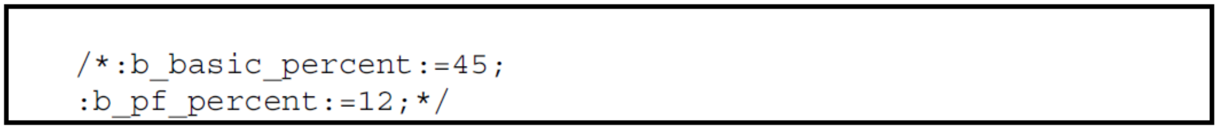
Edit the above script as follows:

a) Use single-line comment syntax to comment the lines that create the bind variables, and turn on SERVEROUTPUT.



```
-- VARIABLE b_basic_percent NUMBER
-- VARIABLE b_pf_percent NUMBER
SET SERVEROUTPUT ON
```

b) Use multiple-line comments in the executable section to comment the lines that assign values to the bind variables.



```

   /*:b_basic_percent:=45;
   :b_pf_percent:=12;*/
```

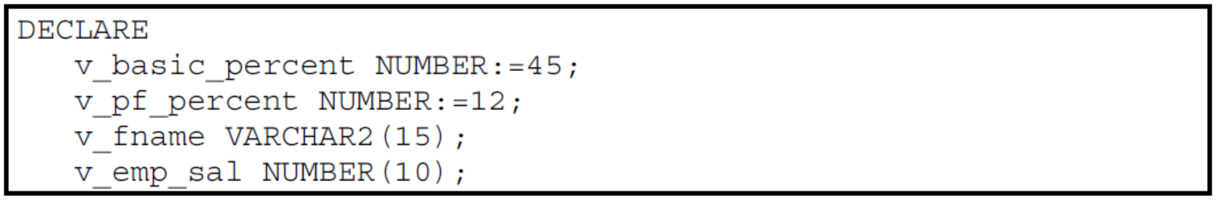
PRACTICE 2:[2 Points]

Use the above script and edit it as follows:

In the declaration section:

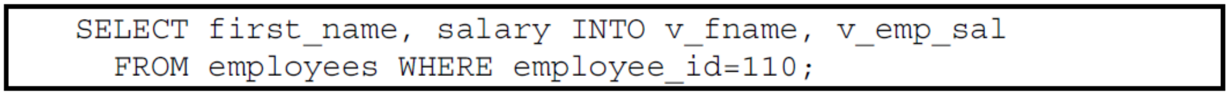
1. Declare and initialize two temporary variables to replace the commented out bind variables

2. Declare two additional variables: v_fname of type VARCHAR2 and size 15, and v_emp_sal of type NUMBER and size 10



```
DECLARE
   v_basic_percent NUMBER:=45;
   v_pf_percent NUMBER:=12;
   v_fname VARCHAR2(15);
   v_emp_sal NUMBER(10);
```

d) Include the following SQL statement in the executable section:



```
   SELECT first_name, salary INTO v_fname, v_emp_sal
     FROM employees WHERE employee_id=110;
```

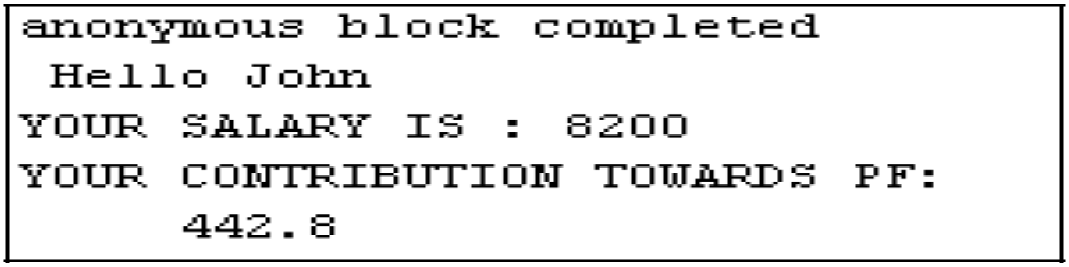
e) Change the line that prints .Hello World. to print .Hello. and the first name. Then, comment the lines that display the dates and print the bind variables.

```
DBMS_OUTPUT.PUT_LINE(' Hello '|| v_fname);
/*   DBMS_OUTPUT.PUT_LINE('TODAY IS : '|| v_today);
DBMS_OUTPUT.PUT_LINE('TOMORROW IS : ' || v_tomorrow);*/
...
...

/
--PRINT b_basic_percent
--PRINT b_basic_percent
```

f) Calculate the contribution of the employee towards provident fund (PF). PF is 12% of the basic salary, and the basic salary is 45% of the salary. Use local variables for the calculation. Try to use only one expression to calculate the PF.
Print the employee's salary and his or her contribution toward PF.

```
   DBMS_OUTPUT.PUT_LINE('YOUR SALARY IS : '||v_emp_sal);
   DBMS_OUTPUT.PUT_LINE('YOUR CONTRIBUTION TOWARDS PF:
     '||v_emp_sal*v_basic_percent/100*v_pf_percent/100);
END;
```

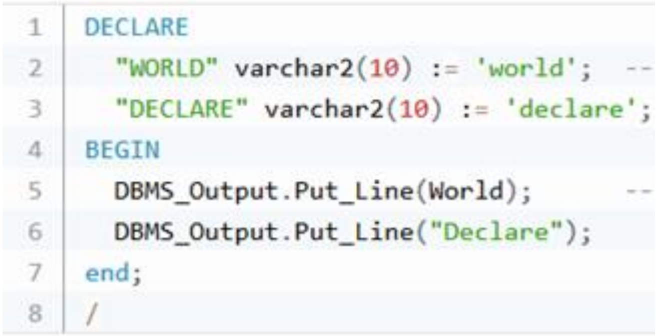
g) Execute and save your script as lab_03_03_soln.sql. The sample output is as follows:



```
anonymous block completed
 Hello John
YOUR SALARY IS : 8200
YOUR CONTRIBUTION TOWARDS PF:
        442.8
```

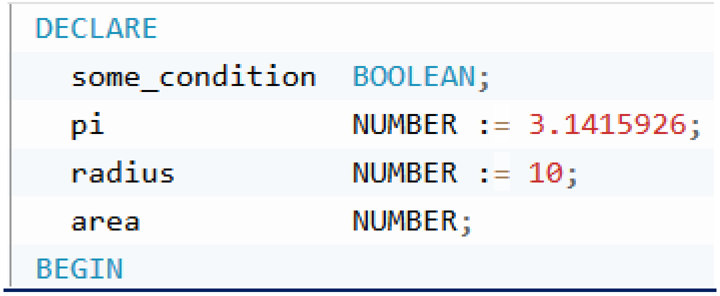
PRACTICE 3:[2 Points]

a) Identify the errors in the following code:



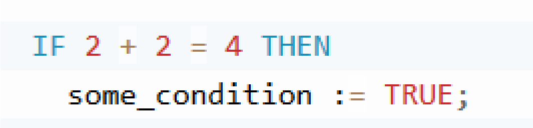
```
DECLARE
  "WORLD" varchar2(10) := 'world';  --
  "DECLARE" varchar2(10) := 'declare';
BEGIN
  DBMS_Output.Put_Line(World);      --
  DBMS_Output.Put_Line("Declare");
end;
/
```

b) Add the single line comment (wherever applicable) to the following block:

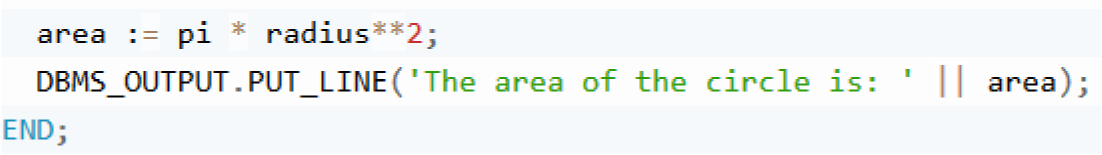


```
DECLARE
  some_condition  BOOLEAN;
  pi              NUMBER := 3.1415926;
  radius          NUMBER := 10;
  area            NUMBER;
BEGIN
```

c) Add the multi line comments(wherever applicable) to the following block. (Execution is not required)

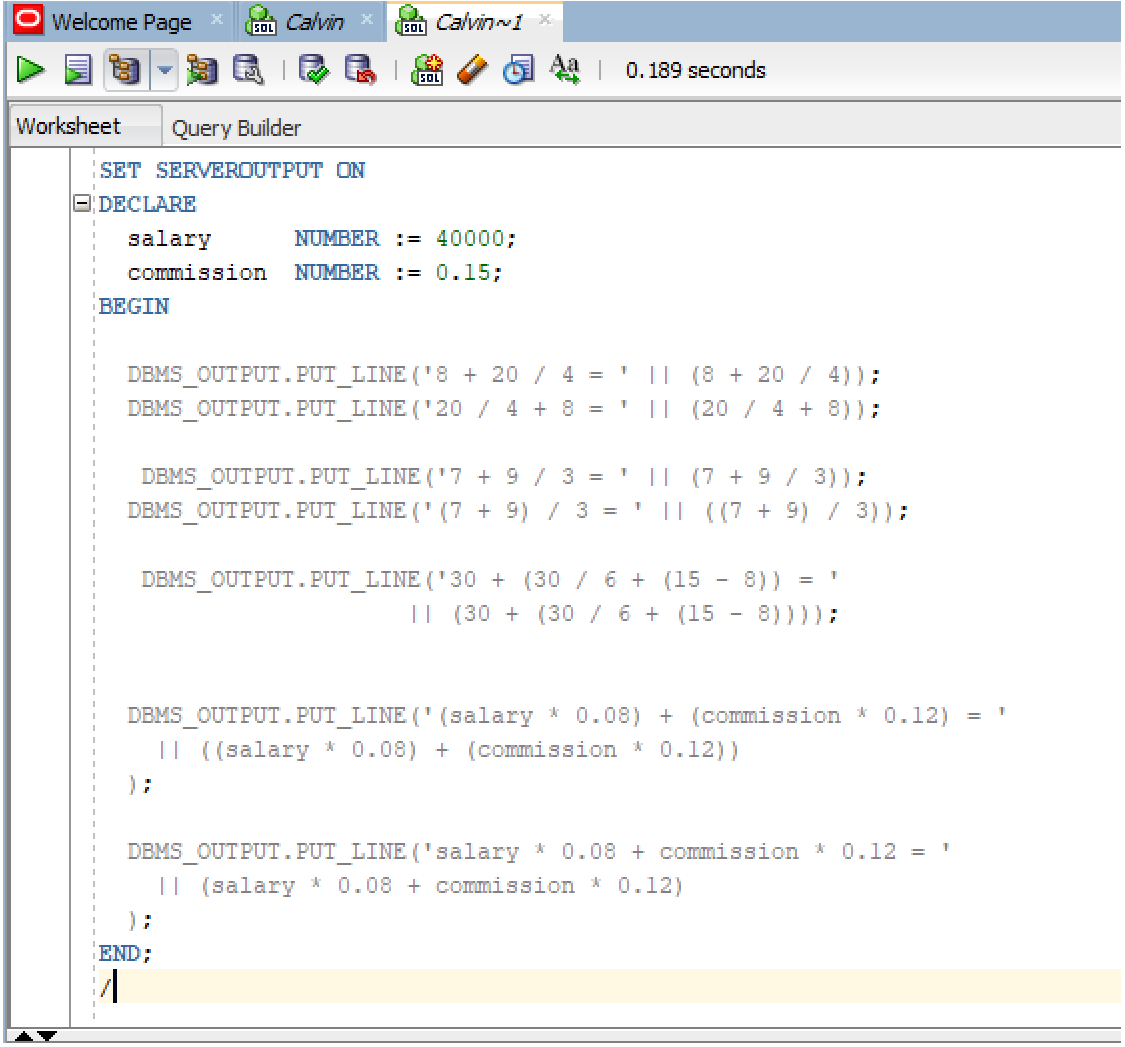


```
  IF 2 + 2 = 4 THEN
    some_condition := TRUE;
```

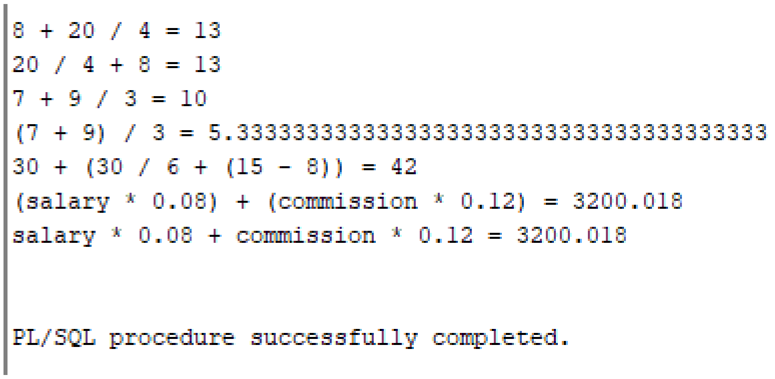


```
  area := pi * radius**2;
  DBMS_OUTPUT.PUT_LINE('The area of the circle is: ' || area);
END;
```

d) Analyze the following code and comment on the operator precedence and parentheses used in the various expressions:



```
SET SERVEROUTPUT ON
DECLARE
  salary      NUMBER := 40000;
  commission  NUMBER := 0.15;
BEGIN

  DBMS_OUTPUT.PUT_LINE('8 + 20 / 4 = ' || (8 + 20 / 4));
  DBMS_OUTPUT.PUT_LINE('20 / 4 + 8 = ' || (20 / 4 + 8));

   DBMS_OUTPUT.PUT_LINE('7 + 9 / 3 = ' || (7 + 9 / 3));
  DBMS_OUTPUT.PUT_LINE('(7 + 9) / 3 = ' || ((7 + 9) / 3));

   DBMS_OUTPUT.PUT_LINE('30 + (30 / 6 + (15 - 8)) = '
                      || (30 + (30 / 6 + (15 - 8))));


  DBMS_OUTPUT.PUT_LINE('(salary * 0.08) + (commission * 0.12) = '
    || ((salary * 0.08) + (commission * 0.12))
  );

  DBMS_OUTPUT.PUT_LINE('salary * 0.08 + commission * 0.12 = '
    || (salary * 0.08 + commission * 0.12)
  );
END;
/
```



```
8 + 20 / 4 = 13
20 / 4 + 8 = 13
7 + 9 / 3 = 10
(7 + 9) / 3 = 5.3333333333333333333333333333333333333333
30 + (30 / 6 + (15 - 8)) = 42
(salary * 0.08) + (commission * 0.12) = 3200.018
salary * 0.08 + commission * 0.12 = 3200.018


PL/SQL procedure successfully completed.
```