# EXPERIMENT 10

**PRE-LAB:**

1. Analyze the code and tell your observation?

# DECLARE

**a number(3) := 100; BEGIN**

**IF (a = 50 ) THEN**

**dbms\_output.put\_line('Value of a is 10' ); ELSEIF ( a = 75 ) THEN**

**dbms\_output.put\_line('Value of a is 20' ); ELSE**

**dbms\_output.put\_line('None of the values is matching'); END IF;**

**dbms\_output.put\_line('Exact value of a is: '|| a ); END;**

1. What will be the output of the following code?

# DECLARE

**lines dbms\_output.chararr; num\_lines number;**

**BEGIN**

**Dbms\_output.enable; dbms\_output.put\_line('Hello!'); dbms\_output.put\_line('Hope you are doing well!'); num\_lines := 2;**

**dbms\_output.get\_lines(lines, num\_lines);**

**FOR i IN 1..num\_lines LOOP dbms\_output.put\_line(lines(i));**

**END LOOP; END;**

1. Consider the following code :−

# DECLARE

**-- Global variables num number := 95; BEGIN**

**dbms\_output.put\_line('num: ' || num1); DECLARE**

**-- Local variables num number := 195; BEGIN**

**dbms\_output.put\_line('num: ' || num1); END;**

**END;**

pg. 39



What will happen when the code is executed?

1. What would be printed when the following code is executed?

# DECLARE

**x NUMBER; BEGIN**

**x := 5;**

**x := 10;**

**dbms\_output.put\_line(-x); dbms\_output.put\_line(+x); x := -10;**

**dbms\_output.put\_line(-x); dbms\_output.put\_line(+x);**

**END;**

1. What will be printed by the following PL/SQL block?

# DECLARE

**a number; b number; c number;**

**PROCEDURE findMin(x IN number, y IN number, z OUT number) IS BEGIN**

**IF x < y THEN**

**z:= x;**

**ELSE**

**z:= y;**

**END IF;**

**END;**

**BEGIN**

**a:= 2;**

**b:= 5;**

**findMin(a, b, c); dbms\_output.put\_line(c);**

**END;**

1. What will be printed by the following PL/SQL block?

# DECLARE

**a number;**

**PROCEDURE squareNum(x IN OUT number) IS BEGIN**

**x := x \* x;**

**END;**

**BEGIN**

**a:= 5;**

**squareNum(a); dbms\_output.put\_line(a);**

**END;**

pg. 40



1. When is the pre-defined exception “**CASE\_NOT\_FOUND**” raised?

# IN-LAB:

Implement PL/SQL Queries on Case Study 9 (MILITARY DATABASE)

1. Create a function that takes solider ID FROM NAVY and return the number of mission done by the soldier in the past.
2. Create a procedure to insert record into army table
3. Create a procedure to delete airforce soldier of age above 50

pg. 41



# POST-LAB:

1. Create a user-defined exception by the name of exp\_check. Select the ename and hiredate of all employees into a cursor. Your program should calculate the experience of all the employees in years, and insert the ename and experience of each employee into temp table. If any employee has experience less than 2 years, the program should be aborted with a suitable message. Raise the user-defined exception exp\_check to achieve this. Display the results on the screen using dbms\_output.put\_line.
2. Create a table EMPLOYEE with the following columns:-

Employee No. Varchar2 4 Employee Name Varchar2 30 Designation Varchar2 10

Category Character 1 Basic Salary Number 4

Category may be ?J?, ?S?, or ?W? for Jr. officers, Sr. officers or Worker category. Formulae:-

DA = 35% of Basic Salary correct up to paise.

HRA = 15% of Basic Salary subject to a maximum of Rs. 250/1000/30000 for categories W/J/S

respectively.

Gross = Basic Salary +DA +HRA

Output the Employee Number and the Gross for each employee in a separate table.

CREATE TABLE EMPLOYEE ( "Employee No " Varchar2 (4), "Employee Name" Varchar2 (30), Designation Varchar2 (10), Category Character (1), "Basic Salary " Number (4) )

--Insert dummy data:-

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| --Empl | EmployeeName | DESIGNATION | C | Basic Salary |
| --1000 | rakesh | Sr.off | s | 3000 |
| --1001 | peeyoosh | Sr.off | s | 4000 |
| --1002 | malik | Jr.off |  | s 5000 |

CREATE TABLE employee\_gross

CREATE TABLE gross ( "Employee No " Varchar2 (4), "Gross Salary " Number (4) )

1. Write a program to read in a number and print it out digit by digit, as a series of words. For example, the number 523 would be printed as "five two three". Use decode function within a for loop. Display the results on the screen using dbms\_output.put\_line.

pg. 42