1)

SET SERVEROUTPUT ON

SET VERIFY OFF

ACCEPT temptype PROMPT 'Enter your input scale (C or F) for temperature: ';

ACCEPT tempvalue PROMPT 'Enter your temperature value to be converted: ';

DECLARE

v\_temptype CHAR(1) := '&temptype';

v\_tempvalue DECIMAL(4,1) := '&tempvalue';

BEGIN

IF UPPER(v\_temptype) = 'C' THEN

v\_tempvalue := v\_tempvalue \* 9 / 5 + 32;

DBMS\_OUTPUT.PUT\_LINE('Your converted temperature in F is exactly ' || v\_tempvalue);

ELSIF UPPER(v\_temptype) = 'F' THEN

v\_tempvalue := (v\_tempvalue - 32) / 1.8;

DBMS\_OUTPUT.PUT\_LINE('Your converted temperature in C is exactly ' || v\_tempvalue);

ELSE

DBMS\_OUTPUT.PUT\_LINE('This is NOT a valid scale. Must be C or F.');

END IF;

END;

/

Output:

anonymous block completed

This is NOT a valid scale. Must be C or F.

anonymous block completed

Your converted temperature in F is exactly 86

anonymous block completed

Your converted temperature in C is exactly -31.7

2)

SET SERVEROUTPUT ON

SET VERIFY OFF

ACCEPT istid PROMPT 'Please enter the Instructor Id: ';

DECLARE

v\_istid NUMBER(3) := &istid;

v\_seccount NUMBER(3);

v\_istname instructor.first\_name%TYPE;

v\_message VARCHAR2(50);

BEGIN

SELECT first\_name || ' ' || last\_name

INTO v\_istname

FROM instructor

WHERE instructor\_id = v\_istid;

SELECT count(section\_id)

into v\_seccount

FROM section s

WHERE instructor\_id = v\_istid;

DBMS\_OUTPUT.PUT\_LINE('Instructor, '|| v\_istname ||', teaches '|| v\_seccount ||' section(s)');

v\_message := CASE

WHEN v\_seccount < 5 THEN 'This instructor may teach more sections. '

WHEN v\_seccount <= 9 THEN 'This instructor may teach more sections. '

WHEN v\_seccount > 9 THEN 'This instructor needs to rest in the next term.'

ELSE

'This instructor teaches enough sections. '

END;

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

EXCEPTION

WHEN NO\_DATA\_FOUND THEN

DBMS\_OUTPUT.PUT\_LINE('This is not a valid instructor');

END;

/

Output:

anonymous block completed

This is not a valid instructor

anonymous block completed

Instructor, Tom Wojick, teaches 10 section(s)

This instructor needs to rest in the next term.

anonymous block completed

Instructor, Fernand Hanks, teaches 9 section(s)

This instructor may teach more sections.

anonymous block completed

Instructor, Rick Chow, teaches 0 section(s)

This instructor may teach more sections.

3)

SET SERVEROUTPUT ON

SET VERIFY OFF

ACCEPT posint PROMPT 'Please enter a Positive Integer: ';

DECLARE

v\_posint NUMBER(5) := &posint;

v\_sum NUMBER(10) := 0;

v\_count NUMBER(5);

v\_msg CHAR(4);

BEGIN

IF v\_posint < 0 THEN

DBMS\_OUTPUT.PUT\_LINE('Not a Positive Interger');

ELSE

IF MOD(v\_posint, 2) = 0 THEN

v\_msg := 'Even';

v\_count := v\_posint;

WHILE v\_count > 0 LOOP

v\_sum := v\_sum + v\_count;

v\_count := v\_count - 2;

END LOOP;

ELSE

v\_msg := 'Odd';

v\_count := v\_posint;

WHILE v\_count > 0 LOOP

v\_sum := v\_sum + v\_count;

v\_count := v\_count - 2;

END LOOP;

END IF;

DBMS\_OUTPUT.PUT\_LINE('The sum of ' || v\_msg ||' integers between 1 and ' || v\_posint || ' is ' || v\_sum );

END IF;

END;

/

Output:

anonymous block completed

The sum of Even integers between 1 and 12 is 42

anonymous block completed

The sum of Odd integers between 1 and 901 is 203401

4)

SET SERVEROUTPUT ON

SET VERIFY OFF

ACCEPT lctid PROMPT 'Enter valid Loction Id:'

DECLARE

v\_lctid departments.location\_id%TYPE := &lctid;

v\_dptidcount NUMBER(5);

v\_empcount NUMBER(5);

i NUMBER(3);

j NUMBER(3);

BEGIN

UPDATE departments SET location\_id = 1400 WHERE department\_id IN (40, 70);

--check how many department in that location

SELECT count(department\_id) INTO v\_dptidcount FROM departments WHERE location\_id = v\_lctid;

--check how many employees in that location based on up's result

SELECT count(last\_name) INTO v\_empcount FROM employees WHERE department\_id in (SELECT department\_id FROM departments WHERE location\_id = v\_lctid);

FOR i IN 1..v\_dptidcount LOOP

DBMS\_OUTPUT.PUT\_LINE('Outer Loop: Department #' || i);

FOR j IN 1..v\_empcount LOOP

DBMS\_OUTPUT.PUT\_LINE('\* Inner Loop: Employee #' || j);

END LOOP;

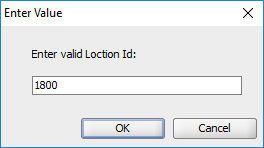
END LOOP;

END;

/

ROLLBACK;

Output:



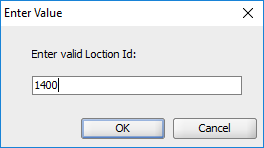
anonymous block completed

Outer Loop: Department #1

\* Inner Loop: Employee #1

\* Inner Loop: Employee #2

rolledback



anonymous block completed

Outer Loop: Department #1

\* Inner Loop: Employee #1

\* Inner Loop: Employee #2

\* Inner Loop: Employee #3

\* Inner Loop: Employee #4

\* Inner Loop: Employee #5

\* Inner Loop: Employee #6

\* Inner Loop: Employee #7

Outer Loop: Department #2

\* Inner Loop: Employee #1

\* Inner Loop: Employee #2

\* Inner Loop: Employee #3

\* Inner Loop: Employee #4

\* Inner Loop: Employee #5

\* Inner Loop: Employee #6

\* Inner Loop: Employee #7

Outer Loop: Department #3

\* Inner Loop: Employee #1

\* Inner Loop: Employee #2

\* Inner Loop: Employee #3

\* Inner Loop: Employee #4

\* Inner Loop: Employee #5

\* Inner Loop: Employee #6

\* Inner Loop: Employee #7

rolledback