

## Nested Blocks and Variable Scope

### Terminology

Directions: Identify the vocabulary word for each definition below.

1. \_\_\_\_\_ Allows clean separation of the error processing code from the executable code so that a program can continue operating in the presence of errors.
2. \_\_\_\_\_ A label given to a block.
3. \_\_\_\_\_ Consists of all the blocks in which the variable is either local (the declaring block) or global (nested blocks within the declaring block) .
4. \_\_\_\_\_ The exception reproduces itself in successive enclosing blocks until a handler is found or there are no more blocks to search in.
5. \_\_\_\_\_ The portion of the program where the variable can be accessed without using a qualifier.

### Try It / Solve It

1. Evaluate the PL/SQL block below and determine the value of each of the following variables according to the rules of scoping.

```
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;
  -- Position 1 --
END;
```

```
weight := weight + 1;
message := message || ' is in stock';
-- Position 2 --
END;
```

- A. The value of weight at position 1 is:
- B. The value of new\_locn at position 1 is:
- C. The value of weight at position 2 is:

- D. The value of message at position 2 is:
- E. The value of new\_locn at position 2 is:
2. Enter and run the following PL/SQL block, which contains a nested block. Look at the output and answer the questions.

```
DECLARE
  v_employee_id employees.employee_id%TYPE;
  v_job         employees.job_id%TYPE;
BEGIN
  SELECT employee_id, job_id INTO v_employee_id, v_job
     FROM employees
     WHERE employee_id = 100;

  DECLARE
    v_employee_id employees.employee_id%TYPE;
    v_job         employees.job_id%TYPE;
  BEGIN
    SELECT employee_id, job_id INTO v_employee_id, v_job
       FROM employees
       WHERE employee_id = 103;
    DBMS_OUTPUT.PUT_LINE(v_employee_id|| ' is a '||v_job);
  END;

  DBMS_OUTPUT.PUT_LINE(v_employee_id|| ' is a '||v_job);
END;
```

- A. Why does the inner block display the job\_id of employee 103, not employee 100?
- B. Why does the outer block display the job\_id of employee 100, not employee 103?
- C. Modify the code to display the details of employee 100 in the inner block. Use block labels.

3. Enter and run the following PL/SQL block. Explain the output. Note: the WHEN OTHERS handler successfully handles any type of exception which occurs.

```
DECLARE
  v_number  NUMBER(2);
BEGIN
  v_number := 9999;
EXCEPTION
  WHEN OTHERS THEN
    DBMS_OUTPUT.PUT_LINE('An exception has occurred');
END;
```

4. Modify the block in question 3 to omit the exception handler, then re-run the block. Explain the output.

5. Enter and run the following code and explain the output.

```
DECLARE
  v_number  NUMBER(4);
BEGIN
  v_number := 1234;

  DECLARE
    v_number  NUMBER(4);
  BEGIN
    v_number := 5678;
    v_number := 'A character string';
  END;

EXCEPTION
  WHEN OTHERS THEN
    DBMS_OUTPUT.PUT_LINE('An exception has occurred');
    DBMS_OUTPUT.PUT_LINE('The number is: '||v_number);
END;
```