

## DES720 Lab 8 – Printouts only incl. Code and Output

### Due by Friday, November 22<sup>nd</sup> by 3pm

1) Walkthrough the following PL/SQL code and try to guess what 3 values will be printed during its execution.

```
SET SERVEROUTPUT ON
<<big>>
DECLARE
    v_mine NUMBER(4) := 500;
    BEGIN
    <<small>>
    DECLARE
        v_mine NUMBER(3) := 700;
    BEGIN
        dbms_output.put_line('Local V_MINE is here: ' || v_mine);
        dbms_output.put_line('Outer V_MINE is here: ' || big.v_mine);
        big.v_mine := v_mine * 2;
    END;
    dbms_output.put_line('Outer V_MINE is here: ' || v_mine);
END;
/
```

2) Write a PL/SQL block

A. That includes declarations for the following variables:

- o A VARCHAR2 datatype that could accept the string 'Introduction to Oracle Database'
- o A NUMBER that may be assigned 123456.78, but not 123456.789 or 1023456.78
- o A CONSTANT that is initialized to the value '704B'
- o A BOOLEAN
- o A DATE data type initialized to one week from today (without the hours)

B. In the body of the block, place a message with values for each of the variables that received an initialization value.

C. In the body of the block, write a code that will perform following tests -- use a nested IF THEN ELSE statement when needed

- o Check if the VARCHAR2 variable you created contains the word "SQL".
- o If it does, then put a message on the screen that provides the name of the course
- o If it does not, then test to see if the CONSTANT you created contains the room number 704B.
- o If it does, then check whether your VARCHAR2 variable has a value and if yes put a message that states the course name and the room name that you've

reached in this logic. Otherwise, put the message “Course is unknown” and the room name as well.

- o If the CONSTANT does not, then put a message on the screen that states that the “Course and location could not be determined”.
- D. Assign the VARCHAR2 variable “C++ advanced” value, just before your IF-THEN logic and observe how the result has changed
- E. Provide BOTH results at the end of this question

3) Perform the following tasks:

- A. Create a table called Lab1\_tab with two columns (Id as numeric and LName as variable character of maximum 20 characters)
- B. Create a sequence called Lab1\_seq that increments by units of 5 and starts with 1.
- C. Write a PL/SQL block that performs the following in this order:
  - a. Declares two variables to hold values for columns of table Lab1\_tab
  - b. The block then inserts into the table the last name of the student that is enrolled in the most classes and his/her last name contains less than 9 characters. Here use a sequence for the Id.
  - c. Then the student with the least enrollments is inserted in the table, use sequence as well.
  - d. Insert the instructor’s last name teaching the least amount of courses if his/her last name does NOT end on “s”. Here do not use the sequence to generate the ID; instead use your first variable.
  - e. Now insert the instructor teaching the most number of courses and use the sequence to populate his/her Id.
  - f. Save your changes and display the content of your table Lab1\_tab
  - g. You will need to plan for the Exception when more than one student is enrolled in the Maximum (Minimum) number of courses. Same for the Instructor teaching Maximum (Minimum) number of courses. The message stored in the table in that case should be “Multiple Names” (instead of the last name).

Hint: You may use ONE outer block and FOUR INNER blocks (each of them with its EXCEPTION section) followed by the appropriate INSERT statement.