Kabeer Ahmed
SE-19028

Database Management System (DBMS – 204)

Experiment # 12

Student Name: Kabeer Ahmed

Roll Number: SE-19028

Maximum Marks	Performance = 05	Viva = 05	Total = 10
Marks Obtained			
Remarks (if any)			

Experiment evaluated by

Instructor Name: Engr. Adiba Jafar

Signature and Date:

Kabeer Ahmed
SE-19028

-Writing Executable Statements

Data Type Conversion

Conversion functions:

– TO_CHAR

– TO_DATE

- **TO_NUMBER**

```
DECLARE
v_date DATE := TO_DATE('12-JAN-2001', 'DD-MON-YYYY');
BEGIN
   –  . . .
```

Kabeer Ahmed

SE-19028

LAB #12
Writing Executable Statements

PRACTICE TASKS

1. **In the executable section, initialize the tomorrow variable with an expression that calculates tomorrow's date (add 1 to the value in today). Print the value of today and tomorrow after printing 'Hello World'.**

```
DECLARE
        v_today DATE := SYSDATE;
        v_tomorow v_today%TYPE := SYSDATE +1;
BEGIN
        DBMS_OUTPUT.PUT_LINE('Hello World');
        DBMS_OUTPUT.PUT_LINE(v_today);
        DBMS_OUTPUT.PUT_LINE(v_tomorow);
END;
/
```

2. **Examine the following code and then answer the questions.**

```
DECLARE
        x VARCHAR2(20);
BEGIN
        x:= '123' + '456' ;
        DBMS_OUTPUT.PUT_LINE(x);
END;
/
```

A. Now, run the code. What is the output?
Ans: **579**

3. **Examine the following code and then answer the questions.**

```
DECLARE
        x NUMBER(6);
BEGIN
        x := 5 + 3 * 2 ;
        DBMS_OUTPUT.PUT_LINE(x);
END;
/
```

A. What do you think the output will be when you run the above code?
Ans: **11**

B. Now run the code. What is the output?
Ans: **11**

C. In your own words, explain the results.

Ans: **Since Precedence of multiplication is more than addition that's why at first 3 and 2 are multiplied and then then result is added in 5, that's why the output is 11.**

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

```
SET SERVEROUTPUT ON
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 –
                DBMS_OUTPUT.PUT_LINE (weight || message ||  new_locn);
        END;
        weight := weight + 1;
        message := message || ' is in stock';
        -- Position 2 –
        DBMS_OUTPUT.PUT_LINE (weight || message || V_NEW_LOCN);
END;
/
```

` A. The value of V_WEIGHT at position 1 is: **2**

B. The value of V_NEW_LOCN at position 1 is: **Western Europe**

C. The value of V_WEIGHT at position 2 is: **601**

D. The value of V_MESSAGE at position 2 is: **Product 10012 is in stock**

E. The value of V_NEW_LOCN at position 2 is: 'V_NEW_LOCN' is not declared

5. **Suppose you created a sub block within a block, as shown above. You declare two variables, V_CUSTOMER and V_CREDIT_RATING, in the main block. You also declare two variables, V_CUSTOMER and V_NAME, in the sub block. Determine the values and data types for each of the following cases mention below:**

```
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_customer v_name v_credit_rating
        END;
        v_customer v_name v_credit_rating
END;
```

```
/
```

A. The value of V_CUSTOMER in the sub block is: **201.**
B. The value of V_NAME in the sub block is: **Unisports.**
C. The value of V_CREDIT_RATING in the sub block is: **EXCELLENT.**
D. The value of V_CUSTOMER in the main block is: **Womansport.**
E. The value of V_NAME in the main block is: ‘V_NAME’ is not declared.
F. The value of V_CREDIT_RATING in the main block is: **EXCELLENT.**

6. **Create and execute a PL/SQL block that accepts two numbers through iSQL*Plus substitution variables. The first number should be divided by the second number and have the second number added to the result. The result should be stored in a PL/SQL variable and printed on the screen.**

a. Use the DEFINE command to provide the two values.

```
DEFINE p_num1 =2
DEFINE p_num2 =4
```

b. Pass the two values defined in step a above, to the PL/SQL block through iSQL*Plus substitution variables.

Ans:

```
SET SERVEROUTPUT ON
DEFINE p_num1 = 2;
DEFINE p_num2 = 4;
DECLARE
        num1 number(7,3) := &p_num1;
        num2 number(7,3) := &p_num2;
        v_result number(7,3);
BEGIN
        v_result:= num1/num2;
        v_result := v_result + num2;
        dbms_output.put_line('RESULT : ' || v_result);
END;
/
```