Practical no. 2

Aim: Writing a PL/SQL block with basic programming constructs by including the following.

1) Sequential statements
2) Unconstrained loop

1) Sequential Statement

a. Write a pl/sql block to perform arithmetic operation entered by the user.

Program:

```
set serveroutput on

accept operation char prompt "Enter the operation(+, -, *, /): "
accept n1 number prompt "Enter first number: "
accept n2 number prompt "Enter second number: "

DECLARE
    N1 NUMBER;
    N2 NUMBER;
    OPERATION VARCHAR(1);
BEGIN
    N1 := &N1;
    N2 := &N2;
    OPERATION := '&OPERATION';

    IF OPERATION = '+' THEN
        DBMS_OUTPUT.PUT_LINE(chr(10)||'The addition of '||N1||' and
'||N2||' is '||(N1+N2));
    ELSIF OPERATION = '-' THEN
        DBMS_OUTPUT.PUT_LINE('The addition of '||N1||' and '||N2||' is
'||(N1-N2));
    ELSIF OPERATION = '*' THEN
        DBMS_OUTPUT.PUT_LINE('The addition of '||N1||' and '||N2||' is
'||(N1*N2));
    elsif OPERATION = '/' THEN
        DBMS_OUTPUT.PUT_LINE('The addition of '||N1||' and '||N2||' is
'||(N1/N2));
    END if;
END;
/
```

Output:

```
Enter the operation(+, -, *, /): -
Enter first number: 10
Enter second number: 8
old   6:     N1 := &N1;
new   6:     N1 := 10;
```

```
old   7:     N2 := &N2;
new   7:     N2 := 8;
old   8:     OPERATION := '&OPERATION';
new   8:     OPERATION := '-';
The addition of 10 and 8 is 2

Enter the operation(+, -, *, /): /
Enter first number: 100
Enter second number: 2
old   6:     N1 := &N1;
new   6:     N1 := 100;
old   7:     N2 := &N2;
new   7:     N2 := 2;
old   8:     OPERATION := '&OPERATION';
new   8:     OPERATION := '/';
The addition of 100 and 2 is 50
```

2) Unconstrained loop

a. Write a pl/sql block to generate table of 20

Program:

```
set serveroutput on;
accept num number prompt "Enter the number: ";

DECLARE
    num number;
    i number;
BEGIN
    num := &num;
    i := 1;
    DBMS_OUTPUT.PUT_LINE(chr(10));

    loop
        DBMS_OUTPUT.PUT_LINE(num||' * '||i||' = '||num*i);
        i:=i+1;
        exit when i > 10;
    end loop;
end;
/
```

Output:

```
Enter the number: 20
old   5:     num := &num;
new   5:     num :=  20;

20 * 1 = 20
20 * 2 = 40
20 * 3 = 60
20 * 4 = 80
20 * 5 = 100
```

```
20 * 6 = 120
20 * 7 = 140
20 * 8 = 160
20 * 9 = 180
20 * 10 = 200
```

b. To show the number between 1000-1010

Program:

```
set serveroutput on

accept num1 number prompt "Enter the first number: "
accept num2 number prompt "Enter the second number: "

DECLARE
    NUM1 NUMBER;
    NUM2 NUMBER;

BEGIN
    num1 := &num1;
    num2 := &num2;
    DBMS_OUTPUT.PUT_LINE(CHR(10));

    LOOP
        DBMS_OUTPUT.PUT_LINE(NUM1);
        NUM1 := NUM1+1;
        EXIT WHEN NUM1>NUM2;
    END LOOP;
END;
/
```

Output:

```
Enter the first number: 1000
Enter the second number: 1010
old   6:     num1 := &num1;
new   6:     num1 := 1000;
old   7:     num2 := &num2;
new   7:     num2 := 1010;

1000
1001
1002
1003
1004
1005
1006
1007
1008
1009
1010
```