Experiment 1.3

Name: Rohan Jaiswal	**UID:** 21BCS2856
Branch: CSE	**Section/Group:** 608 (B)
Semester: 3rd	**Date of Performance:** 08/09/2022
Subject Name: DBMS	**Subject Code:** 21CSH-214

Aim:

Introduction and implementation of programs using Block Structure and variables. PL/SQL.

Features :

1. It is a Block Structural Language consists of Procedural statement and SQL statements.
2. Whole block of code can be executed by oracle engine in one go.
3. We can make use of variable for processing and holding the result and also update table with values.
4. Can create function or procedure and can be called repeatedly.

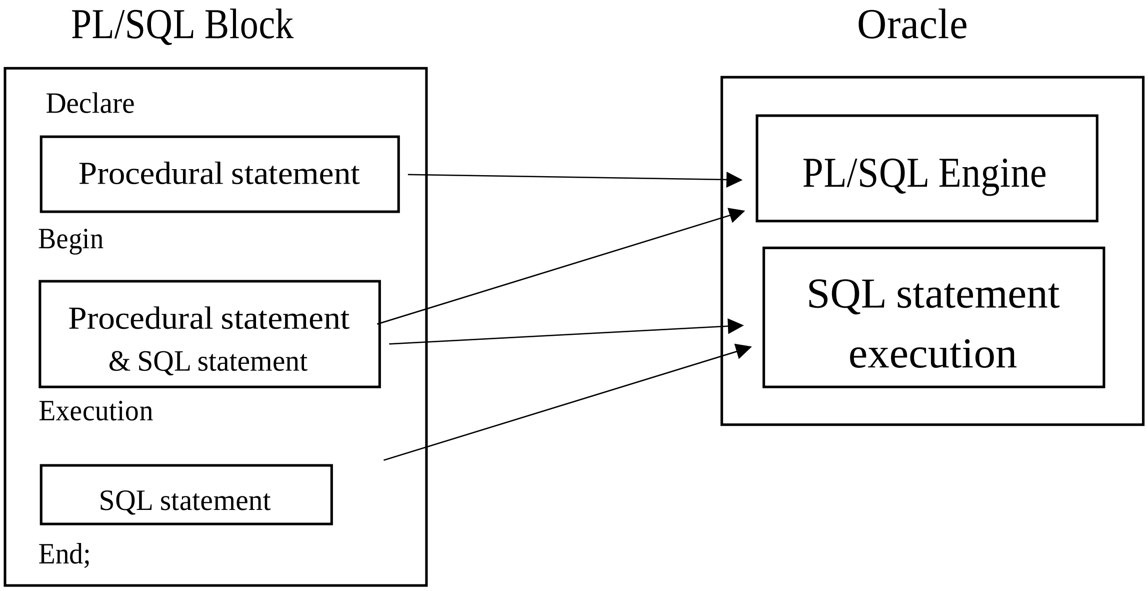
Components:

1. Declare Section: Declaration / Initialization variables both can be done in this section.
2. Begin Section : Contains executable and processing statement for manipulation of data
3. Exception Section : Contains error handling codes.
4. End Section : Indicates the end of PL/SQL.

Simple Template of PL/SQL Block :

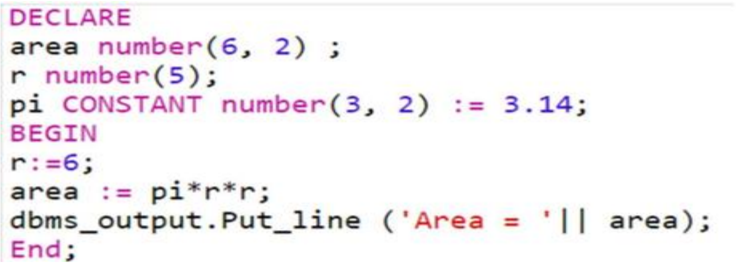
```
Declare
        < Variable declaration statements;
Begin
        < Execution statements;
Exception
        < Error handling statements;
End;
```

Execution environment in PL/SQL



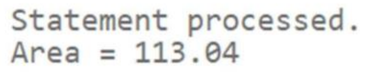
DBMS script and Output:

1. Write PL/SQL command to find the area of circle.



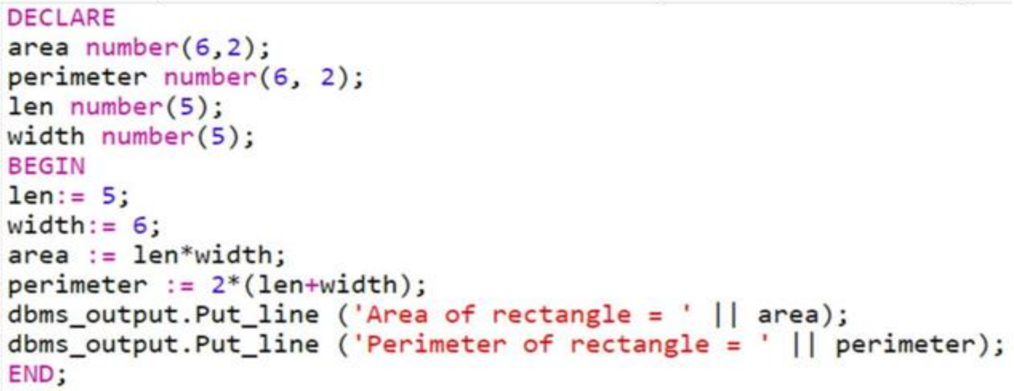
```
DECLARE
area number(6, 2) ;
r number(5);
pi CONSTANT number(3, 2) := 3.14;
BEGIN
r:=6;
area := pi*r*r;
dbms_output.Put_line ('Area = '|| area);
End;
```

Output:



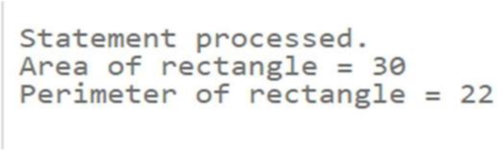
```
Statement processed.
Area = 113.04
```

2. Write PL/SQL command to find the area and perimeter of Rectangle.



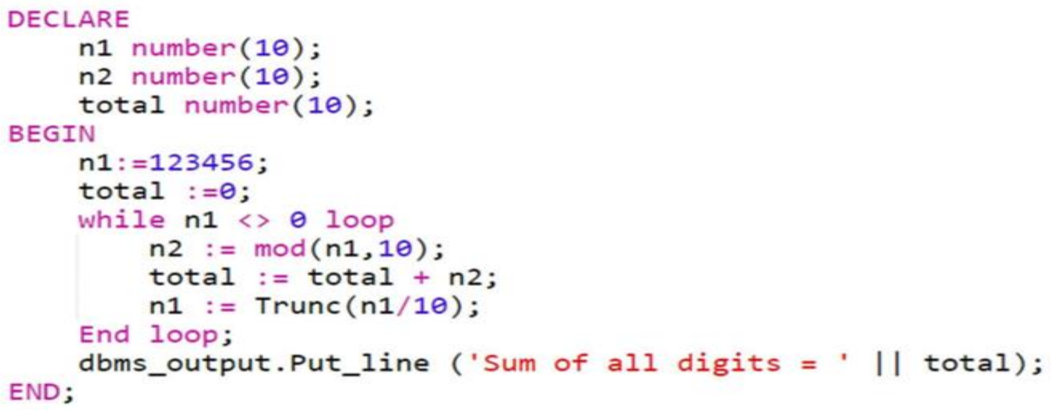
```
DECLARE
area number(6,2);
perimeter number(6, 2);
len number(5);
width number(5);
BEGIN
len:= 5;
width:= 6;
area := len*width;
perimeter := 2*(len+width);
dbms_output.Put_line ('Area of rectangle = ' || area);
dbms_output.Put_line ('Perimeter of rectangle = ' || perimeter);
END;
```

Output:



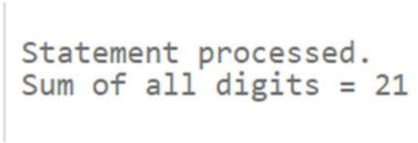
```
Statement processed.
Area of rectangle = 30
Perimeter of rectangle = 22
```

3. Write PL/SQL command to find sum of digit of given number.



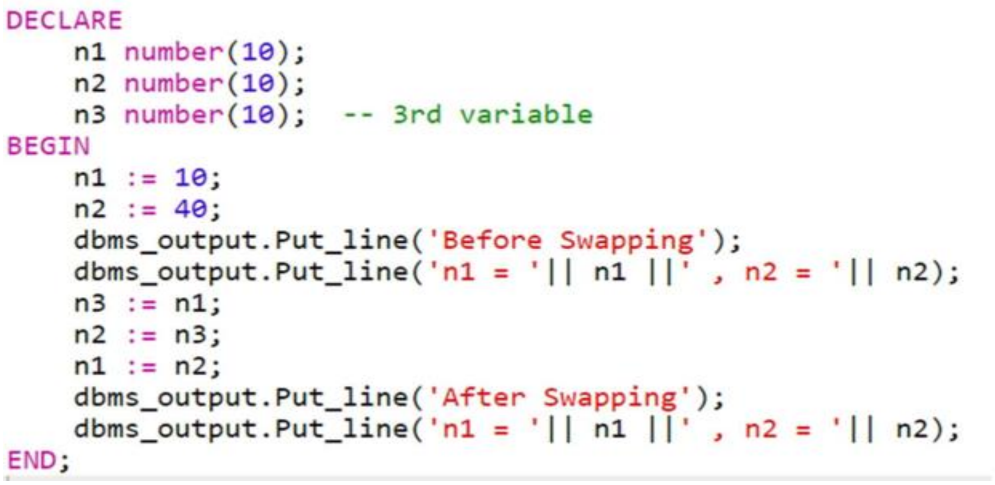
```
DECLARE
     n1 number(10);
     n2 number(10);
     total number(10);
BEGIN
     n1:=123456;
     total :=0;
     while n1 <> 0 loop
          n2 := mod(n1,10);
          total := total + n2;
          n1 := Trunc(n1/10);
     End loop;
     dbms_output.Put_line ('Sum of all digits = ' || total);
END;
```

Output:



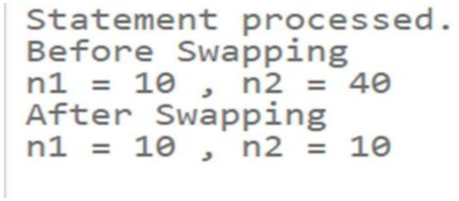
```
Statement processed.
Sum of all digits = 21
```

4. Write PL/SQL command to Swap two variable with and without third variable. I. With Third variable



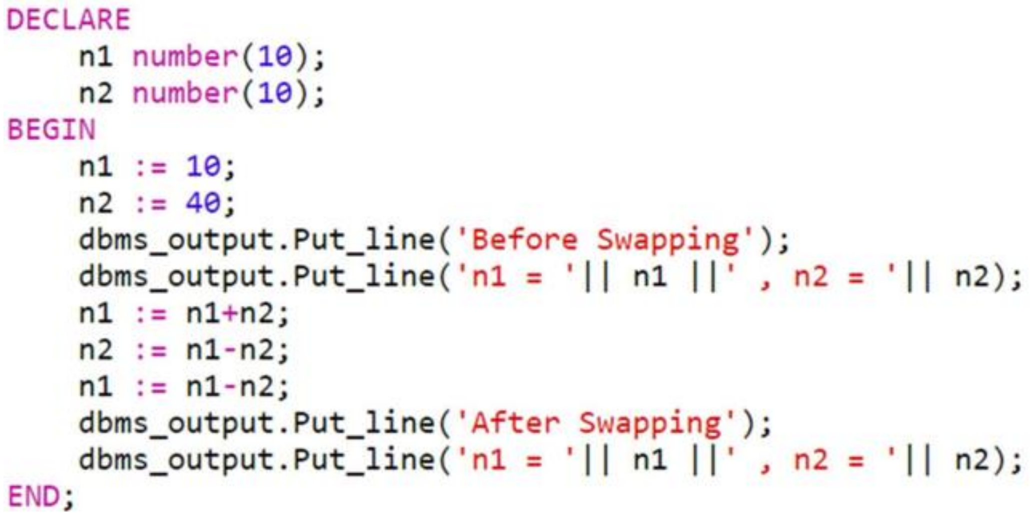
```
DECLARE
     n1 number(10);
     n2 number(10);
     n3 number(10);   -- 3rd variable
BEGIN
     n1 := 10;
     n2 := 40;
     dbms_output.Put_line('Before Swapping');
     dbms_output.Put_line('n1 = '|| n1 ||' , n2 = '|| n2);
     n3 := n1;
     n2 := n3;
     n1 := n2;
     dbms_output.Put_line('After Swapping');
     dbms_output.Put_line('n1 = '|| n1 ||' , n2 = '|| n2);
END;
```

Output:



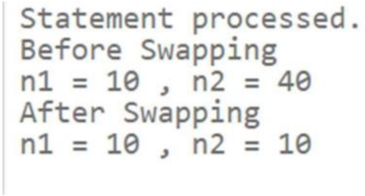
```
Statement processed.
Before Swapping
n1 = 10 , n2 = 40
After Swapping
n1 = 10 , n2 = 10
```

II. Without Third variable



```
DECLARE
    n1 number(10);
    n2 number(10);
BEGIN
    n1 := 10;
    n2 := 40;
    dbms_output.Put_line('Before Swapping');
    dbms_output.Put_line('n1 = '|| n1 ||' , n2 = '|| n2);
    n1 := n1+n2;
    n2 := n1-n2;
    n1 := n1-n2;
    dbms_output.Put_line('After Swapping');
    dbms_output.Put_line('n1 = '|| n1 ||' , n2 = '|| n2);
END;
```

Output:



```
Statement processed.
Before Swapping
n1 = 10 , n2 = 40
After Swapping
n1 = 10 , n2 = 10
```

Learning outcomes (What I have learnt):

1. Learned about DBMS languages.
2. I have learned about PL/SQL block Structure.
3. Learn about Four components of Pl/SQL and their function.