EXPERIMENT NO. 1

Write PL/SQL code block to display the word “Hello!”

PROGRAM

```
set serveroutput on
BEGIN
  dbms_output.put_line('Hello!');
END;
```

Output:

```
Hello!
PL/SQL procedure successfully completed.
```

EXPERIMENT NO. 2

Write PL/SQL code block which will get the salary of an employee with particular id from emp table and display it on screen.

PROGRAM

```
create table emp_detail(id number, name char(30), salary number(6));
insert into emp_detail values(101, 'sargam', 25000);
insert into emp_detail values(102, 'harman', 30000);
insert into emp_detail values(103, 'sargun', 50000);
insert into emp_detail values(104, 'sehaj', 27000);
insert into emp_detail values(105, 'shetal', 45000);
insert into emp_detail values(106, 'aman', 65000);
insert into emp_detail values(107, 'simar', 45000);
insert into emp_detail values(108, 'samreen', 44000);
insert into emp_detail values(109, 'ishjot', 33000);
select * from emp_detail;
```

ID	NAME	SALARY
101	sargam	25000
102	harman	30000
103	sargun	50000
104	sehaj	27000
105	shetal	45000
106	aman	65000
107	simar	45000
108	samreen	44000
109	ishjot	33000

Method 1:

```
set serveroutput on
DECLARE
  emp_rec emp_detail%rowtype;
BEGIN
   select * into emp_rec from emp_detail where id = 104;
   dbms_output.put_line('Employee ID: ' || emp_rec.id);
   dbms_output.put_line('Employee Name: ' || emp_rec.name);
   dbms_output.put_line('Employee Salary: ' || emp_rec.salary);
END;
```

Output:

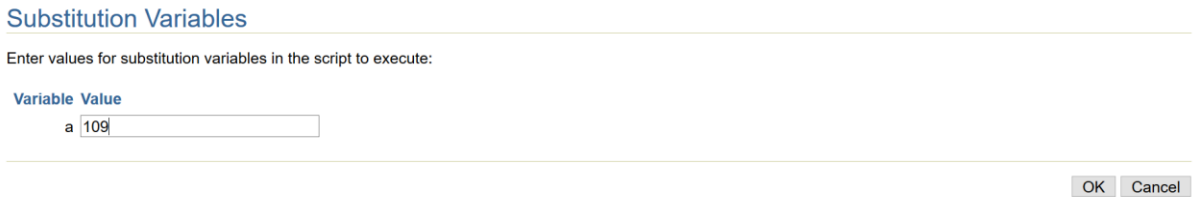
```
Employee ID: 104
Employee Name: sehaj
Employee Salary: 27000
PL/SQL procedure successfully completed.
```

Method 2:

```
set serveroutput on
DECLARE
  x varchar(10);
```

```
  a varchar(10);
BEGIN
   select salary into x from emp_detail where id = '&a';
   dbms_output.put_line(x);
END;
```

Output:




```
old 5: select salary into x from emp_detail where id = '&a';
new 5: select salary into x from emp_detail where id = '109';
33000
PL/SQL procedure successfully completed.
```

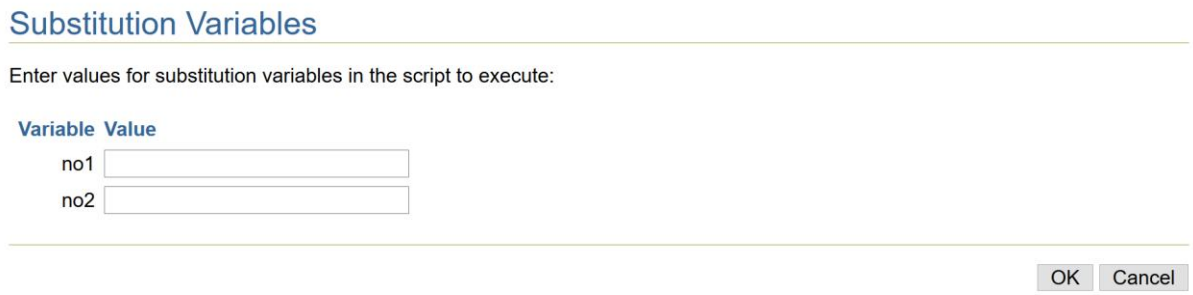
EXPERIMENT NO. 3

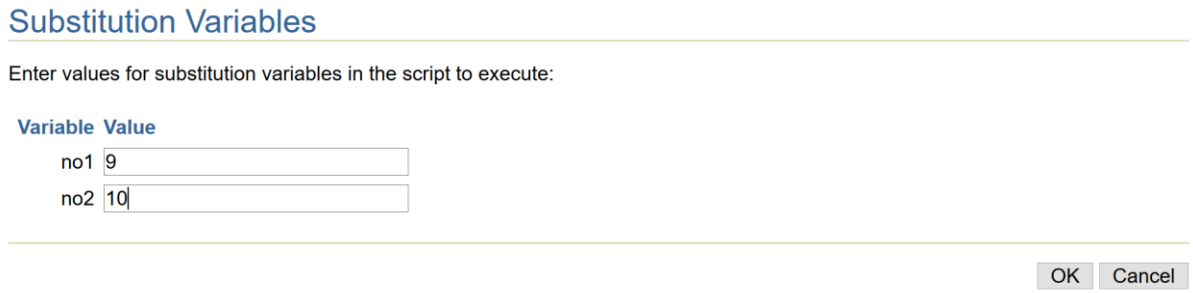
Write PL/SQL code block to calculate the sum by taking input of two numbers.

PROGRAM

```
set serveroutput on
DECLARE
   no1 number(7);
   no2 number(7);
   result number(7);
BEGIN
   result:= &no1 + &no2;
   dbms_output.put_line('Sum of the numbers entered is: '||result);
END;
```

Output:






```
old 6: result:= &no1 + &no2;
new 6: result:= 9 + 10;
Sum of the numbers entered is: 19
PL/SQL procedure successfully completed.
```

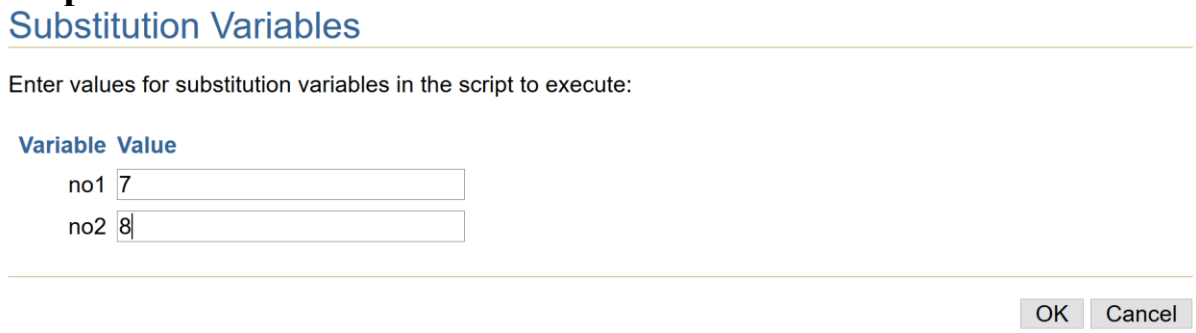
EXPERIMENT NO. 4

Write PL/SQL code block which creates two variables in the outer block and assign their product to the third variable created in the inner block.

PROGRAM

```
set serverouput on
DECLARE
   no1 number(7);
   no2 number(7);
BEGIN
   no1 := &no1;
   no2 := &no2;
   DECLARE
      result number(15);
   BEGIN
      result := no1 * no2;
      dbms_output.put_line('The product is: '||result);
   END;
END;
```

Output:




```
old 5: no1 := &no1;
new 5: no1 := 7;
old 6: no2 := &no2;
new 6: no2 := 8;
The product is: 56
PL/SQL procedure successfully completed.
```

EXPERIMENT NO. 5

Write PL/SQL code block to declare a record called employee_rec based on user-defined datatype.

PROGRAM

```
set serveroutput on
DECLARE
type employee is record
( id number(4),
name varchar(30),
salary number(6));
employee_rec employee;
BEGIN
select id, name, salary into employee_rec.id, employee_rec.name,
employee_rec.salary from emp_detail where id = 109;
dbms_output.put_line('Employee ID: ' || employee_rec.id);
dbms_output.put_line('Employee Name: ' || employee_rec.name);
dbms_output.put_line('Employee Salary: ' || employee_rec.salary);
END;
```

Output:

```
Employee ID: 109
Employee Name: ishjot
Employee Salary: 33000
PL/SQL procedure successfully completed.
```

EXPERIMENT NO. 6

Write PL/SQL code block for a procedure that has four sections. Each section should output a statement use labels and goto command to output the section message in the following order.

Section 3
Section 2
Section 1
Section 4

PROGRAM

```
set serveroutput on

BEGIN
   dbms_output.put_line('Gursimar Kaur');
   dbms_output.put_line('1820036');
   goto section3;

   <<section1>>
   dbms_output.put_line('Section 1');
   goto section2;

   <<section2>>
   dbms_output.put_line('Section 2');
   goto section4;

   <<section3>>
   dbms_output.put_line('Section 3');
   goto section1;

   <<section4>>
   dbms_output.put_line('Section 4');

END;
```

Output:

```
Gursimar Kaur
1820036
Section 3
Section 1
Section 2
Section 4
PL/SQL procedure successfully completed.
```

EXPERIMENT NO. 7

Write PL/SQL code block which use the relational operators to compare character values for equality or inequality.

PROGRAM

```
set serveroutput on
DECLARE
Name varchar(20);
Company varchar(30);
Introduction CLOB;
Choice varchar(1);
BEGIN
   Name := 'Gursimar';
   Company := 'Google';
   Introduction := 'Smart and talented';
   choice := '&choice';
   if choice = 'y' then
      dbms_output.put_line(Name);
      dbms_output.put_line(Company);
      dbms_output.put_line(Introduction);
   else
      dbms_output.put_line('wrong input');
   end if;
END;
```

Output:

```
old 10: choice := '&choice';
new 10: choice := 'y';
Gursimar
Google
Smart and talented
PL/SQL procedure successfully completed.
```

EXPERIMENT NO. 8

Write PL/SQL code block which use the if statement.

PROGRAM

```
set serveroutput on
DECLARE
no1 number(4);
no2 number(4);
no3 number(4);
min_no number(4);
Procedure find_min(x In number, y In number, z In number, min Out
number) IS
BEGIN
if x<y then
   if x<z then
      min := x;
   else
      min := z;
   end if;
else
   if y<z then
      min := y;
   else
      min := z;
   end if;
end if;
end find_min;
BEGIN
find_min(&a, &b, &c, min_no);
dbms_output.put_line('Minimum value among all 3 numbers is :
'||min_no);
END;
```

```
old 23: find_min(&a, &b, &c, min_no);
new 23: find_min(3, 4, 2, min_no);
Minimum value among all 3 numbers is : 2
PL/SQL procedure successfully completed.
```