Output

Hello World

Date. _____

Expt. 1 Page No. 1

Write a program in PL/SQL to print Hello World on screen.

```
SET SERVEROUTPUT ON;
DECLARE
BEGIN
  dbms_output.put_line ('HELLO World');
END;
```

Teacher's Sign. _____

Output

ACC_NO	Branch	Type	OPR_Mode	Ver_Emp_NO	CURBAL	Status
1001	A1	SB	S1	E1	5000	A

Date.

Expt. 2

Page No. 2

Write a PL/SQL query to check and update balance by deducting fine if balance is less than minimum balance

```
SET SERVEROUTPUT ON;
DECLARE
    min_bal number(5) := 6000;
    mACC_NO varchar(15);
    mFINE number(4) := 100;
    mCUR_BAL number(11);
BEGIN
    mACC_NO := &acc_no;
    SELECT curbal INTO mCUR_BAL FROM ACCT_MSTR
    WHERE ACC_NO = mACC_NO;
    IF mCUR_BAL < MIN_BAL THEN
        UPDATE ACCT_MSTR set curbal = curbal - mFINE
        WHERE ACC_NO = mACC_NO;
    END IF;
END;
```

Teacher's Sign.

Output	
Radius	Area
3	28.26
4	50.24
5	78.5
6	113.04
7	153.86

Date. _____

Expt. 3.

Page No. 3

Write a PL/SQL query for area of circle using simple loop

```
SET SERVEROUTPUT ON;
CREATE TABLE AREAS(Radius number(5), Area number(5));
DECLARE
   r number := 3;
   a number;
BEGIN
   LOOP
      a := 3.14 * r * r;
      INSERT INTO AREAS VALUES(r, a);
      r := r + 1;
      EXIT WHEN r > 7;
   END LOOP;
END;
```

Teacher's Sign. _____

Output	
Side	Area
3	9
4	16
5	25
6	36
7	49

Date. _____

Expt. 4 Page No. 4

Write a PL/SQL query for area of square using while loop.

```
SET SERVEROUTPUT ON;
CREATE TABLE AREAS( Side number (5), Area number(5));
DECLARE
   side number := 3;
   a number;
BEGIN
   WHILE side <=7
      LOOP
      a := side * side;
      INSERT INTO AREAS VALUES (side, a);
      side := side +1;
   END LOOP;
END;
```

Teacher's Sign. _____

Output
Inverted number is 9365

Date. _____

Expt. 5 Page No. 5

Write a PL/SQL query to reverse a number using for loop?

```
SET SERVEROUTPUT ON;
DECLARE
  given_number varchar(5) := '5639';
  str_length number(2);
  inverted_number varchar(5);
BEGIN
  str_length := length(given_number);
  FOR cntr IN REVERSE 1..str_length
  LOOP
    inverted_number := inverted_number || substr(given_
    number, cntr, 1)
  END LOOP;
  dbms_output.put_line('Inverted number is '||inverted_number);
END;
```

Teacher's Sign. _____

Output
3 ABC 7000

Date. _____

Expt. 6 a)

Page No. 6

Write a query which will get salary of a particular employee with id from Emp table and display it on screen.

```
Declare
        vId     varchar(5) := 3;
        vName   Employee.Name%type;
        vSal    Employee.Salary%type;
Begin
    Select Name, Salary Into vName, vSal from
    Employee wher Id = vID;
    dbms_output.put_line(vID||' '||vName||' '||vSal);
END;
```

Teacher's Sign. _____

Output

21

Date. _____

Expt. 6 b) Page No. 7

Write a query which will create two variables in outer block and assign their product to third variable in inner block.

```
Declare
    num1  number(2) := 7;
    num2  number(2) := 8;
Begin
    Declare
        num3  number(2);
    Begin
        num3 := num1 * num2;
        dbms_output.put_line(num3);
    END;
END;
```

Teacher's Sign. _____

Output

YES table updated, incentive = 10500.
YES table updated, incentive = 6750.

Date.

Expt. 7

Page No. 8

Write a PL/SQL procedure to calculate the incentive on a target achieved and display message either the record updated or not.

```
Declare Procedure Incentive(Salary number, Quantity
number, Employee ID number) IS
incentive number := 0;
updated varchar(3) := 'NO';
Begin
  If Salary > (Quantity + 200) then
    incentive := (Salary - Quantity)/4;
  Updat Employee set Salary = Salary + incentive
  Where Employee_Id = Employee ID;
  updated := 'YES';
  END IF;
  dbms_output.put_line(updated || 'table updated, incentive='
                       || incentive);
  END Incentive;
Begin
   Incentive(50000, 8000, 2);
   Incentive(36000, 9000, 1);
END;
```

Teacher's Sign.

Output

The employees are in the department 1 : 4
There are some vacancies in department 1

Date. _____

Expt. 8 Page No. 9

Write PL/SQL code to count number of employee in particular department and check whether this department have any vacancies or not. Assume there are 45 vacancies in department

```
Declare
    total_emp number;
    x1 varchar(5);
Function check_vacancy (x IS number)
Return varchar IS
    result varchar(5) := 'NO'
Begin
    Select count(*) into total_emp From Employee e
    Join Department d on e.E_Id = d.Dept_Id where
    e.Dept_Id = x;
    dbms_output.put_line('The employees are in the
                          department'||x||':'||To_Char(totalemp));
    If total_emp > 45 then
      dbms_output.put_line('There are no vacancies in
                            department'||x);
    Else
      dbms_output.put_line('There are some vacancies in
                            department'||x);
      result := 'YES';
    END If;
    Return Result;
END;
Begin
    x1 := check_vacancy(1);
END;
```

Teacher's Sign. _____

Output
UNKNOWN
NO
YES

Date.

Expt. 9 Page No. 10

Write a PL/SQL procedure to accepts a boolean and use case to print Unknown, Yes & No based on input.

```
Create OR Replace Procedure take_boolean(bool BOOLEAN)
AS
BEGIN
    CASE
        WHEN bool IS NULL Then dbms_output.put_line('UNKNOWN');
        WHEN bool Then dbms_output.put_line('YES');
        WHEN NOT bool Then dbms_output.put_line('NO');
    END CASE;
END;

BEGIN
    take_boolean(NULL);
    take_boolean(FALSE);
    take_boolean(TRUE);
END;
```

Teacher's Sign.

Output

The employee DEF works in salary 15500 which is higher than mid-range of salary 13000.

Date.

Expt. 10

Page No. 11

Write a program to update salary of employee by 8% if salary exceeds the mid range of salary against job and update upto mid range if salary is less than mid range, and display a suitable message.

```
Declare
  emp_min_salary number(5);
  emp_max_salary number(5);
  emp_mid_salary number(5);
  tmp_salary EMPLOYEES.SALARY%TYPE;
  tmp_emp_id EMPLOYEES.EMPLOYEE_ID%TYPE := 1002;
  tmp_emp_name EMPLOYEES.FIRST_NAME%TYPE;

BEGIN
  Select MIN_SALARY, MAX_SALARY INTO emp_min_salary,
  emp_max_salary From JOBS Where JOB_ID = (
  Select JOB_ID From EMPLOYEES Where EMPLOYEES
  Where EMPLOYEE_ID = tmp_emp_id);

  emp_mid_salary := (emp_min_salary + emp_max_salary)/2;

  Select SALARY, FIRST_NAME INTO tmp_salary, tmp_emp_name
  From EMPLOYEES Where EMPLOYEE_ID = tmp_emp_id;

  If tmp_salary < emp_mid_salary Then
    UPDATE EMPLOYEES SET SALARY = emp_mid_salary Where
    EMPLOYEE_ID = tmp_emp_id;
```

Teacher's Sign.

```
ELSE
    UPDATE EMPLOYEES SET SALARY = SALARY + SALARY *
    8/100 where EMPLOYEE_ID = tmp_emp_id;
END If;

If tmp_salary > emp mid_salary Then
    dbms_output.put_line('The employee' || tmp_emp_name ||
    'works in salary' || TO_CHAR(tmp_salary) || 'which is
    higher than mid-range of salary' || TO_CHAR(emp_mid_salary));
ELSE If tmp_salary < emp_mid_salary Then
    dbms_output.put_line('The employee' || tmp_emp_name ||
    'works in salary' || TO_CHAR(tmp_salary) || 'which is
    lower than mid-range of salary' || TO_CHAR(emp_mid_salary));
ELSE
    dbms_output.put_line('The employee' || tmp_emp_name ||
    'works in salary' || TO_CHAR(tmp_salary) || 'which is
    equal to mid-range of salary' || TO_CHAR(emp_mid_salary));
END IF;
END;
```

Teacher's Sign. _____

Output

```
The First 10 numbers are:
1
4
7
10
13
16
19
22
25
28
```

Date. _____

Expt. 11 a) Page No. 13

Write a program in PL/SQL to print 1st n numbers with a difference of 3 and starting from 1, which uses For loop to insert ten rows into a table.

```
Create Table Inserted_values(Numbers number(5));

Declare
 n number := 10;
 i number := 1;
 m number := 1;
Begin
 dbms_output.put_line('The first '||n||' numbers
                                are: ');
 dbms_output.put_line(i||' ');
 Insert Into Inserted_values values(i);
 For i In 1..n-1 Loop
    m := m+3;
    dbms_output.put_line(m||' ');
    Insert Into Inserted_values(m);
 END LOOP;
END;
```

Teacher's Sign. _____

Out Put

```
The first 10 numbers are:
1
4
7
10
13
16
19
22
25
28
```

Date.

Expt. 11 b)

Page No. 14

To demostrate the use of While loop

```
Declare
  n number := 10;
  i number := 1;
  m number := 1;
Begin
  dbms_output.put_line('The first '||n||' numbers
                                    are: ');
  dbms_output.put_line(i);
  While i<10 Loop
     m:=m+3;
     dbms_output.put_line(m);
     i:=i+1;
  END Loop;
  END;
```

Teacher's Sign.

Output

```
Entering  Outer  loop
Entering  Inner  loop
Exiting   Inner  loop
Exiting   Outer  loop
```

Expt. 11 c)

Date.

Page No. 15

To demonstrate the use of 'Nested loop'

```
Declare
    i number;
    j number;
Begin
    For i In 1..1 loop
        dbms_output.put_line('Entering Outer loop');
        For j In 1..1 Loop
            dbms_output.put_line('Entering Inner loop');
            dbms_output.put_line('Exiting Inner loop');
        End loop;
        dbms_output.put_line('Exiting Outer loop');
    End loop;
End;
```

Teacher's Sign.

Output

```
i is: 1 and j is: 1
i is: 1 and j is: 2
i is: 2 and j is: 1
i is: 2 and j is: 2
```

Date.

Expt. 11 d)

Page No. 16

To demonstrate the use of 'Labeling loop'

```
Declare
    i number;
    j number;
Begin
    <<Outer loop>>
    For i In 1..2 loop
        <<Inner loop>>
        For j In 1..2 loop
            dbms_output.put_line('i is: '||i||' and j is: '||j);
        End loop Inner_loop;
    End loop Outer_loop;
End;
```

Teacher's Sign.

Output

```
Value of a : 10
Value of a : 11
Value of a : 12
Value of a : 13
Value of a : 14
Value of a : 16
Value of a : 17
Value of a : 18
Value of a : 19
```

Date. _____

Expt. 11e) Page No. 17

To demonstrate the use of GOTO statement.

```
Declare
    a number := 10;
Begin
    << loop start >>
    while a < 20 loop
    dbms_output.put_line('Value of a:' || a);
    a := a + 1;
    If a = 15 Then
        a := a + 1;
        GOTO loop start;
    End If;
  End loop;
End;
```

Teacher's Sign. _____

Output

```
Employee : Bill
Salary : 40000
----------

Employee : Will
Salary : 30000
----------

Employee : Smith
Salary : 27000
----------

Employee : Mark
Salary : 26000
----------

Employee : Dustin
Salary : 25000
----------
```

Date.

Expt. 12 a)

Page No. 18

Write a program in PL/SQL which uses a cursor to select the five highest paid employees from the emp table.

```
Declare
   Cursor EMP IS Select EMP_Name, Salary From
   EMP_Cur Order By Salary DESC;

Begin
   For i In EMP
   Loop
      If EMP%rowcount <= 5 Then
         dbms_output.put_line('Employee: '||i.EMP_Name);
         dbms_output.put_line('Salary: '||i.Salary);
         dbms_output.put_line('----------');
      End If;
   End loop;
End;
```

Teacher's Sign.

Output
Employees Updated.

Date. _____

Expt. 12 b) Page No. 19

To use implicit cursor attributes to update the salary of employees in emp table.

```
Begin
  Update Employees Set Salary = Salary + 1300;
  If sql%notfound Then
     dbms_output.put_line('No Employees Updated');
  ELSIF sql%found Then
     dbms_output.put_line('Employees Updated');
  End If;
END;
```

Teacher's Sign. _____

Output

```
Mike    20000
Dustin  25000
Mark    26000
Smith   27000
Will    30000
Bill    40000
```

Date.

Expt. 12 c)

Page No. 90

To illustrate the use of different types of explicit cursors.

```
Declare
    e_name EMP_Cur.EMP_Name%type;
    e_sal EMP_Cur.Salary%type;
    Cursor c_employees Is Select EMP_Name, Salary
    From EMP_Cur;

Begin
    Open c_employees;
    Loop
        Fetch c_employees Into e_name, e_sal;
        Exit When c_employees%notfound;
        dbms_output.put_line(e_name||' '||e_salary);
    End loop;
    Close c_employees;
End;
```

Teacher's Sign.

Output

```
Old Salary : 34300
New Salary : 34800
Salary Difference : 500
```

Date.

Expt. 12 d)

Page No. 21

To illustrate the use of Triggers.

```
Create Trigger display_salary_changes
Before Delete OR Insert OR Update On Employees
For Each Row
When (New.Employee_Id > 0)
Declare
    sal_difference number;
Begin
    sal_difference := :New.salary - :Old.salary;
    dbms_output.put_line('Old Salary: ' || :Old.salary);
    dbms_output.put_line('New Salary: ' || :New.salary);
    dbms_output.put_line('Salary Difference: ' || sal_difference);
End;
```

```
Update Employees Set Salary = Salary + 500 Where
Employee_Id = 1003;
```

Teacher's Sign.

Before Calling

Output		
101	Mike	20000
102	Dustin	25000

After Calling

Output		
101	Mike	20000
103	Will	27000

Expt. 12 e)

Date.

Page No. 92

To illustrate the use of Packages.

```
Create Package Body emp_package As
Procedure addEmployee(e_id EMP_Cur.E_id%type, e_name
 EMP_Cur.EMP_Name%type, e_sal EMP_Cur.Salary%type)
Is
Begin
   Insert Into EMP_Cur Values(e_id, e_name, e_sal);
End addEmployee;

Procedure delEmployee(e_id EMP_Cur.E_Id%type) Is
Begin
   Delete From EMP_Cur Where EMP_Id = e_Id;
End delEmployee;
END emp_package;

Begin
   emp_package.addEmployee(103, 'Will', 27000);
   emp_package.delEmployee(102);
End;
```

Teacher's Sign.

Output
Error !

Expt. 12 f)

Date.

Page No. 23

To handle Exceptions

```
Declare
    e_id EMP_Cur.E_Id%type := 104;
    e_name EMP_Cur.EMP_Name%type;
    e_sal EMP_Cur.Salary%type;

Begin
    Select EMP_Name, Salary Into e_name, e_sal
    From EMP_Cur Where E_id = e_id;
    dbms_output.put_line('Name: '|| e_name);
    dbms_output.put_line('Salary: '|| e_sal);
Exception
    When no_data_found Then
        dbms_output.put_line('No Such Employee!');
    When others Then
        dbms_output.put_line('Error!');
END;
```

Teacher's Sign.