Date ___/___/___

1. Write a program to print 'Hello'.
set serveroutput on

```
declare
begin
    dbms_output.put_line('Hello');
exception
 end;
/
```

2. Write a program to print addition of two numbers.

```
declare
a number := 10;
b number := 20;
c number;
begin
 c:=a+b;
 dbms_output.put_line('Addition is'||c);
 end;
/
```

3. Subraction and multiplication of two numbers with user choice value.
User difine function :-

```
declare
     a number := &a;
     b number := &b;
     c number ;
```

```
begin
   c:= a-b;
 dbms_output.put_line ('subtraction is'||c);
  c:=a* b ;
 dbms_output.put_line ('Multiplication is'||c);
 end;
 /
```

4. Assiging (faching) values from table
 (first create table)

emp	
emp no.	emp name
111	Ayush
222	Radha

```
 declare
 no emp.emp_no%type;
name emp.ename %type;
 begin
 select emp_no, ename into no, name from
 emp where emp_no=222;
 dbms_output.put_line ('Employee name is'||
                                     name);
 end;
 /
```

5. find a squre of a number.

```
 declare
  a number;
  b number;
begin
  a:=5;
  b:=a*a;
  dbms_output.put_line ('square is'|| b);
end;
/
```

6. Find a cube of a given number.

```
declare
    a number := &a;
    b number;
   begin
   b := a*a* a ;
   dbms_ output . put_line ('cube is' || b);
   end;
  /
```

7. Write a program to calculate the AREA and store that value in the table AREAS (RADIUS NUMBER(5), AREA NUMBER (14,2)).

```
declare
PI constant number (9,7) = 3.1415927;
Radius number (5);
Area number (14 ,2);
begin
   radius := 3;
   AREA := PI* POWER (RADIUS ,2);
Insert into areas values (Radius, Area);
 dbms_output.put_line('AREA is' || AREA);
end;
/
```

8. Find out a simple intrest.

```
declare
    P number := &P;
    r number(3,1) := &r;
    n number := &n;
    si number (7,2);
```

```
begin.
   si:=(p*r*n)/100;
  dbms_output.put_line ('simple interest is'||si);
end;
/
```

9. Write a program to read a number from user and determine whether it is odd or even.

```
declare
        no number := &no;
begin
     If   MOD(no,2)=0 THEN
    dbms_output.put_line('Given number'||no||
                                        'is even');
    ELSE
    dbms_output.put_line('Given number'||no||
                                        'is odd');
    END If;
  end;
 /
```

10. Check number is positive, Negative or zero;

```
declare
        no   number := &no;
begin
      if   no>0  then
    dbms_output.put_line (no||'is positive');
    else  no<0  then
    dbms_output.put_line (no||'is negative');
    else
    dbms_output.put_line(no,||'is zero');
   end if;
  end;
/
```

11. Check number is greatest :-

```
declare
        a number;
        b number;
        c number;
begin
      a:= &a;
      b:= &b;
      c:= &c;
 if a>b and a>c then
  dbms_output.put_line('a is greatest');
 elsif b>a and b>c then
  dbms_output.put_line('b is greatest');
 else
  dbms_output.put_line('c is greatest');
  end if;
end;
/
```

12. Display number 1 to 5 along with their square value using loop construct.

```
declare
      counter number(3):= 1;
```

```
begin
   dbms_output.put_line ('value' ||'square');
   Loop
  EXIT when counter > 5;
 dbms_output.put_line (counter||' '||
                                        counter* counter);
  counter := (counter +1;
  END loop;
end;
/
```

13. Display even number from 1 to 100.

```
declare
        counter   NUMBER(3) := 2;
begin
        loop
        Exit when cointer  >100;
     dbms_output.put_line (counter);
     counter := counter +2;
     END Loop;
     end;
 /
```

15. -- for loop (reverse order):-

```
declare
        counter   Number (3);
begin
        for counter   in   1...10
        Loop
```

```
   dbms_output.put_line (counter);
   END loop;
end;
/
```

14. -- while loop.

```
Declare
     Counter number (3):= 2;
begin
    while counter <50;
    Loop
 dbms_output.put_line (Counter);
  counter := Counter +2 :
  END Loop;
end;
/
```

16. Fetching table records using for loop.

```
begin
  For i in (select * from emp)
  loop
dbms_output.put_line('employee number is :' ||
                                  i.emp_no);
dbms_output.put_line ('employee name is :' ||
                                  i.ename);
  end loop;
  end;
/
```

17. Charge : useqence useing GOTO statement.

```
begin
  dbms_output.put_line ('code starts');
  dbms-output.put-line('Before GOTO statement..');
 GOTO Jump;
 dbms-output.put-line ('This statement will not
                          get executed...');
  << jump >>
  dbms_output.put_line('flow of execution
                              jumped here...');
  end;
/
```

18. Program for breaking a loop.

```
begin
    for i in 1...100
    loop
    if i>50 then
    goto jump;
    end if;
  dbms_output.put_line (i);
   end loop;
 << jump >>
  dbms - output.put_line ('now stop printing...');
end;
/
```

19. Farinhit to salsiys formula.

```
declare
        F number (5,1);
        C number (5,1);
begin
      F := &f;
      C := (F-32)/1.8;
  dbms_output.put_line(c));
  end;
 /
```

20. Deleting a record of given emp id.

```
declare
        eid    emp.emp_id%type;
begin
        eid := &eid;
        delete from emp where emp_id = eid;
  end;
 /
```