FOR LOOP

```
DECLARE
I NUMBER(6) :=1;
BEGIN
FOR J IN REVERSE 1..100 LOOP
DBMS_OUTPUT.PUT_LINE(J||' ');
END LOOP;
END;
/
```

FUNCTIONS

```
create or replace function adder(n1 in number, n2 in number)
return number
is
n3 number(8);
begin
n3 :=n1+n2;
return n3;
end;
/


DECLARE
   n3 number(2);
BEGIN
   n3 := adder(11,22);
   dbms_output.put_line('Addition is: ' || n3);
END;
/
```

GOTO

```
DECLARE
I NUMBER(10) := 1;

BEGIN
<<L1>>
DBMS_OUTPUT.PUT_LINE(I);
I:=I+1;
IF I!=10 THEN
GOTO L1;
END IF;



END;
/
```

IF THEN ELSE

```
declare
percent numeric;
BEGIN
percent :='60';
if percent > 30 then
dbms_output.put_line('PASS');
else
dbms_output.put_line('FAIL');
end if;
end;
/
```

IF THEN

```
declare
num1 number(6);
num2 number(6);
begin
num1 := 16;
num2 := 61;
if num1 < num2 then
dbms_output.put_line(num1||'is less than'||num2);
end if;
end;
/
```

NESTED IF THEN ELSE

```
declare
percent numeric;
BEGIN
percent :='83';
if percent >=75 then
dbms_output.put_line('DISINCTION');
elsif percent>=60 and percent<75 then
dbms_output.put_line('FIRST CLASS');
elsif percent>=50 and percent<60 then
dbms_output.put_line('SECOND CLASS');
elsif percent>=40 and percent<50 then
dbms_output.put_line('PASS CLASS');
else
dbms_output.put_line('FAIL...');
end if;
end;
/
```

PROCEDURE

```
CREATE OR REPLACE PROCEDURE INSERTEMP(ID IN NUMBER,NAME IN CHAR,SAL IN NUMBER)
IS
BEGIN
INSERT INTO EMP VALUES(ID,NAME,SAL);
END;
/

BEGIN
INSERTEMP(200,'HASHIM',50000);
DBMS_OUTPUT.PUT_LINE('INSERTED');
END;
/
```

SARCHED CASE

```
DECLARE
day VARCHAR(10):= TO_CHAR(SYSDATE, 'dd');
today varchar(10);
BEGIN
CASE
WHEN day = 1 THEN
 today :='Monday';
WHEN day = 2 THEN
 today :='Tuesday';
WHEN day = 3 THEN
 today :='Wednesday';
WHEN day = 9 THEN
 today :='Thursday';
WHEN day = 5 THEN
 today :='Friday';
WHEN day = 6 THEN
 today :='Saturday';
WHEN day = 7 THEN
 today :='Sunday';
ELSE
 today :='No such grade';
END CASE;
DBMS_OUTPUT.PUT_LINE(today);
END;
/
```

SIMPLE CASE

```
declare
marks number(10) := 190;
begin
case marks
when 90 then
dbms_output.put_line('First class');

when 35 then
dbms_output.put_line('pass');

else
dbms_output.put_line('failed');

end case;
end;
/
```

TRIGGER

```
CREATE TABLE MEDICAL(ID INT,NAME VARCHAR(20),QUANTITY INT);

INSERT INTO MEDICAL VALUES(1,'TELMA',20);

INSERT INTO MEDICAL VALUES(2,'VICKS',30);

SELECT * FROM MEDICAL;


CREATE TABLE STOCK(ID INT,QUANTITY INT);

CREATE OR REPLACE TRIGGER ITEM
AFTER UPDATE
ON MEDICAL
BEGIN
IF (NEW.QUANTITY <10) THEN
INSERT INTO STOCK VALUES(NEW.ID,100);
END IF;
END;
/
```

WHILE LOOP

```
DECLARE
I NUMBER(6) :=1;
BEGIN
WHILE I<=10
LOOP
DBMS_OUTPUT.PUT_LINE(I||' ');
I:=I+1;
END LOOP;
END;
/
```

```
ZERO DIVIDE

DECLARE
I NUMBER(10) ;

BEGIN
I:= 78/0;
EXCEPTION
WHEN ZERO_DIVIDE THEN
DBMS_OUTPUT.PUT_LINE('DIVIDE BY ZERO ERROR');
END;
/
```