1. DECLARE

v\_num NUMBER:= &sv\_user\_num;

BEGIN

IF MOD(v\_num,2)=0 THEN

DBMS\_OUTPUT.PUT\_LINE(v\_num||'is even number');

ELSE

DBMS\_OUTPUT.PUT\_LINE(v\_num||'is ODD number');

end if;

end;

DECLARE

v\_sal NUMBER;

BEGIN

select sal into v\_sal from emp where empno=&sv\_empno;

IF v\_sal>5000 THEN

DBMS\_OUTPUT.PUT\_LINE('Employee is getting good salary');

ELSE

DBMS\_OUTPUT.PUT\_LINE('Employee is getting avg salary');

end if;

END;

/

DECLARE

v\_date DATE:= TO\_DATE('&sv\_user\_date', 'DD-MON-YYYY');

v\_day VARCHAR2(1);

BEGIN

v\_day:= TO\_CHAR(v\_date, 'D');

CASE v\_day

WHEN '1' THEN

DBMS\_OUTPUT.PUT\_LINE('Today is Sunday');

WHEN '2' THEN

DBMS\_OUTPUT.PUT\_LINE('Today is Monday');

WHEN '3' THEN

DBMS\_OUTPUT.PUT\_LINE('Today is Tueday');

WHEN '4' THEN

DBMS\_OUTPUT.PUT\_LINE('Today is Wednesday');

WHEN '5' THEN

DBMS\_OUTPUT.PUT\_LINE('Today is Thursday');

WHEN '6' THEN

DBMS\_OUTPUT.PUT\_LINE('Today is Friday');

ELSE

DBMS\_OUTPUT.PUT\_LINE('Today is Satday');

END CASE;

END;

DECLARE

v\_sal number;

BEGIN

select sal into v\_sal from emp where empno=&sv\_empno;

CASE WHEN v\_sal>10000 THEN

DBMS\_OUTPUT.PUT\_LINE('Excellent');

WHEN v\_sal>6000 THEN

DBMS\_OUTPUT.PUT\_LINE('Good');

WHEN v\_sal>4000 THEN

DBMS\_OUTPUT.PUT\_LINE('Average');

else

DBMS\_OUTPUT.PUT\_LINE('Poor');

end case;

END;

/

DECLARE

v\_counter BINARY\_INTEGER:=0;

BEGIN

LOOP

-- increment loop counter by one

v\_counter:= v\_counter + 1;

DBMS\_OUTPUT.PUT\_LINE('v\_counter = '||v\_counter);

-- if EXIT condition yields TRUE exit the loop

IF v\_counter = 5 THEN

EXIT; (EXIT WHEN v\_counter=5)

END IF;

END LOOP;

--control resumes here

DBMS\_OUTPUT.PUT\_LINE('Done');

END;

DECLARE

v\_counter NUMBER := 0;

BEGIN

WHILE v\_counter < 5 LOOP

DBMS\_OUTPUT.PUT\_LINE('v\_counter = '||v\_counter);

v\_counter := v\_counter + 1;

END LOOP;

END;

DECLARE

v\_counter NUMBER := 0;

BEGIN

FOR v\_counter IN 1..5 LOOP

DBMS\_OUTPUT.PUT\_LINE('v\_counter = '||v\_counter);

EXIT WHEN v\_counter=3;

END LOOP;

END;

DECLARE

v\_num1 INTEGER := &sv\_num1;

v\_num2 INTEGER := &sv\_num2;

v\_result integer;

BEGIN

v\_result :=v\_num1/v\_num2; -- gives error of v\_num2=0

DBMS\_OUTPUT.PUT\_LINE('v\_result = '||v\_result);

END;

/

DECLARE

v\_num1 INTEGER := &sv\_num1;

v\_num2 INTEGER := &sv\_num2;

v\_result integer;

BEGIN

v\_result :=v\_num1/v\_num2;

DBMS\_OUTPUT.PUT\_LINE('v\_result = '||v\_result);

EXCEPTION

WHEN ZERO\_DIVIDE THEN

DBMS\_OUTPUT.PUT\_LINE('A number cannot be divided by zero.');

END;

/

DECLARE

v\_firstname varchar2(30);

BEGIN

SELECT FIRSTNAME INTO v\_firstname from employees where employee\_id=&sv\_empno;

DBMS\_OUTPUT.PUT\_LINE(v\_firstname);

EXCEPTION

WHEN no\_data\_found THEN

DBMS\_OUTPUT.PUT\_LINE('No employee found with empno');

END;

/

DECLARE

v\_firstname varchar2(30);

v\_num number;

BEGIN

SELECT FIRSTNAME INTO v\_firstname from employees where employee\_id=&sv\_empno;

DBMS\_OUTPUT.PUT\_LINE(v\_firstname);

v\_num:=10/0;

EXCEPTION

WHEN no\_data\_found THEN

DBMS\_OUTPUT.PUT\_LINE('No employee found with empno');

WHEN zero\_divide – others can also be used THEN

DBMS\_OUTPUT.PUT\_LINE('divided by zero');

END;

--if we provide correct empno then it will print second exception otherwise 1st exception

/

DECLARE

v\_firstname varchar2(30);

v\_num number;

BEGIN

SELECT FIRSTNAME INTO v\_firstname from employees;

EXCEPTION

WHEN too\_many\_rows then

DBMS\_OUTPUT.PUT\_LINE('More values');

END;

/

DECLARE

v\_firstname varchar2(30);

BEGIN

select firstname into v\_firstname from employees;

EXCEPTION

WHEN others then

DBMS\_OUTPUT.PUT\_LINE('Unknown Error');

END;

/

DECLARE

v\_employee\_id number:= &empid;

v\_firstname varchar2(30);

e\_invalidid exception;

BEGIN

if v\_employee\_id < 0 then

raise e\_invalidid;

else

select firstname into v\_firstname from employees where employee\_id= v\_employee\_id;

DBMS\_OUTPUT.PUT\_LINE(v\_firstname);

end if;

EXCEPTION

WHEN e\_invalidid then

DBMS\_OUTPUT.PUT\_LINE('no negative values are allowed');

END;

/

DECLARE

v\_value number:= &sv\_value;

e\_ab10000 exception;

e\_ab6000 exception;

e\_other exception;

BEGIN

if v\_value >10000 then

raise e\_ab10000;

elsif v\_value >6000 then

raise e\_ab6000;

else

raise e\_other;

end if;

EXCEPTION

WHEN e\_ab10000 then

DBMS\_OUTPUT.PUT\_LINE('above 10000');

WHEN e\_ab6000 then

DBMS\_OUTPUT.PUT\_LINE('above 6000');

WHEN e\_other then

DBMS\_OUTPUT.PUT\_LINE('other exception');

END;

/

DECLARE

v\_employee\_id number:= &sv\_emp\_id;

v\_firstname varchar2(20);

BEGIN

if v\_employee\_id <0 then

RAISE\_APPLICATION\_ERROR(-20001,'id can not be negative');

else

select firstname into v\_firstname from employees where employee\_id= v\_employee\_id;

end if;

END;

/

CREATE OR REPLACE PROCEDURE first\_procedure

AS

BEGIN

dbms\_output.put\_line('This is my first procedure');

END;

Exec first\_procedure

/

SQL> select object\_name, object\_type, status from user\_objects where object\_name

='FIRST\_PROCEDURE';

/

--file 1

CREATE OR REPLACE PROCEDURE first\_procedure(in\_val in number, out\_val out number)

AS

BEGIN

Out\_val:=sqrt(in\_val);

END;

--file 2

DEECLARE

v\_val number;

Begin

First\_procedure(2,v\_val);

Dbms\_output.put\_line('Sqrt is'|| v\_val);

End;

/

CREATE OR REPLACE PROCEDURE fourth\_procedure(inout\_val in out number)

AS

BEGIN

inOut\_val:=sqrt(inout\_val);

END;

DECLARE

v\_val number default 10;

Begin

fourth\_procedure(v\_val);

Dbms\_output.put\_line('Sqrt is'|| v\_val);

End;

/

CREATE OR REPLACE PROCEDURE get\_name(

pno number,

p\_firstname out varchar2, -- length shouldn’t be specified in procedure or function

p\_lastname out varchar2)

AS

Begin

select firstname, lastname into p\_firstname, p\_lastname from employees

where employee\_id=pno;

exception

when no\_data\_found then

Dbms\_output.put\_line('No employee found');

End;

declare

v\_firstname varchar2(20);

v\_lastname varchar2(20);

Begin

get\_name(1234,v\_firstname, v\_lastname);

Dbms\_output.put\_line(v\_firstname||v\_lastname);

end;

/

--FUNCTION

create or replace function get\_power(in\_val1 in number, in\_val2 in number)

return number

as

v\_val number;

Begin

v\_val:=power(in\_val1, in\_val2);

return(v\_val);

end;

declare

v\_value number;

Begin

v\_value:=get\_power(&sv\_val1, &sv\_val2);

DBMS\_OUTPUT.PUT\_LINE(v\_value);

end;

/

select ename, get\_power(sal,2) from emp

/