1) Print "Welcome To PL/SQL Block"

```
BEGIN
DBMS_OUTPUT.PUT_LINE('Welcome To PL/SQL Block');
END;
/
```

2) Count Square

```
DECLARE
n NUMBER := 5;
sq NUMBER;
BEGIN
sq := n * n;
DBMS_OUTPUT.PUT_LINE('Square of ' || n || ' is ' || sq);
END;
/
```

3) Find Simple Interest

```
DECLARE
p NUMBER := 1000;
r NUMBER := 5;
t NUMBER := 2;
si NUMBER;
BEGIN
si := (p * r * t) / 100;
DBMS_OUTPUT.PUT_LINE('Simple Interest = ' || si);
END;
/
```

4) Global and Local Variable

```
DECLARE
g_var NUMBER := 100;
BEGIN
DECLARE
l_var NUMBER := 50;
BEGIN
DBMS_OUTPUT.PUT_LINE('Global Variable: ' || g_var);
DBMS_OUTPUT.PUT_LINE('Local Variable: ' || l_var);
END;
END;
/
```

5) Constant Variable

```
DECLARE
pi CONSTANT NUMBER := 3.14;
BEGIN
DBMS_OUTPUT.PUT_LINE('Value of PI = ' || pi);
END;
/
```

6) Ticket Age Check

```
DECLARE
age NUMBER := 15;
BEGIN
IF age < 5 THEN
DBMS_OUTPUT.PUT_LINE('Free Ticket');
ELSIF age BETWEEN 5 AND 12 THEN
```

```
DBMS_OUTPUT.PUT_LINE('Half Ticket');
ELSE
DBMS_OUTPUT.PUT_LINE('Full Ticket');
END IF;
END;
/
```

7) Maximum and Minimum

```
DECLARE
a NUMBER := 10;
b NUMBER := 25;
c NUMBER := 5;
BEGIN
DBMS_OUTPUT.PUT_LINE('Maximum = ' || GREATEST(a,b,c));
DBMS_OUTPUT.PUT_LINE('Minimum = ' || LEAST(a,b,c));
END;
/
```

8) Student Marks

```
DECLARE
rollno NUMBER := 1;
m1 NUMBER := 70;
m2 NUMBER := 80;
m3 NUMBER := 90;
total NUMBER;
per NUMBER;
grade CHAR(1);
BEGIN
total := m1+m2+m3;
per := total/3;
IF per >= 75 THEN grade := 'A';
ELSIF per >= 60 THEN grade := 'B';
ELSIF per >= 50 THEN grade := 'C';
ELSE grade := 'F';
END IF;
DBMS_OUTPUT.PUT_LINE('Roll No: '|| rollno);
DBMS_OUTPUT.PUT_LINE('Total: '|| total);
DBMS_OUTPUT.PUT_LINE('Percentage: '|| per);
DBMS_OUTPUT.PUT_LINE('Grade: '|| grade);
END;
/
```

9) CASE Statement

```
DECLARE
grade CHAR(1) := 'B';
BEGIN
CASE grade
WHEN 'A' THEN DBMS_OUTPUT.PUT_LINE('Excellent');
WHEN 'B' THEN DBMS_OUTPUT.PUT_LINE('Good');
WHEN 'C' THEN DBMS_OUTPUT.PUT_LINE('Average');
ELSE DBMS_OUTPUT.PUT_LINE('Fail');
END CASE;
END;
/
```

10) First 10 Numbers LOOP EXIT

```
DECLARE
```

```
i NUMBER := 1;
BEGIN
LOOP
DBMS_OUTPUT.PUT_LINE(i);
i := i+1;
EXIT WHEN i > 10;
END LOOP;
END;
/
```

11) First 10 Numbers WHILE

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

12) First 10 Numbers FOR

```
BEGIN
FOR i IN 1..10 LOOP
DBMS_OUTPUT.PUT_LINE(i);
END LOOP;
END;
/
```

13) Reverse Numbers 10 to 1

```
BEGIN
FOR i IN REVERSE 1..10 LOOP
DBMS_OUTPUT.PUT_LINE(i);
END LOOP;
END;
/
```

14) First 10 Prime Numbers

```
DECLARE
n NUMBER := 2;
count NUMBER := 0;
i NUMBER;
flag BOOLEAN;
BEGIN
WHILE count < 10 LOOP
flag := TRUE;
FOR i IN 2..TRUNC(SQRT(n)) LOOP
IF MOD(n,i)=0 THEN
flag := FALSE; EXIT;
END IF;
END LOOP;
IF flag THEN
DBMS_OUTPUT.PUT_LINE(n);
count := count + 1;
END IF;
n := n+1;
END LOOP;
```

```
END;
/
```

15) Reverse a String

```
DECLARE
str VARCHAR2(50) := 'HELLO';
rev VARCHAR2(50) := '';
BEGIN
FOR i IN REVERSE 1..LENGTH(str) LOOP
rev := rev || SUBSTR(str,i,1);
END LOOP;
DBMS_OUTPUT.PUT_LINE('Reverse: ' || rev);
END;
/
```

16) String Pyramid

```
DECLARE
str VARCHAR2(20) := 'PLSQL';
BEGIN
FOR i IN 1..LENGTH(str) LOOP
DBMS_OUTPUT.PUT_LINE(SUBSTR(str,1,i));
END LOOP;
END;
/
```

17) Number Pyramid

```
DECLARE
i NUMBER;
j NUMBER;
BEGIN
FOR i IN 1..5 LOOP
FOR j IN 1..i LOOP
DBMS_OUTPUT.PUT(j || ' ');
END LOOP;
DBMS_OUTPUT.NEW_LINE;
END LOOP;
END;
/
```

18) GOTO Statement

```
DECLARE
i NUMBER := 1;
BEGIN
<<start_loop>>
DBMS_OUTPUT.PUT_LINE(i);
i := i+1;
IF i <= 5 THEN
GOTO start_loop;
END IF;
END;
/
```

19) Insert 10 Rows into Square Table

```
BEGIN
FOR i IN 1..10 LOOP
INSERT INTO square_table VALUES (i, i*i);
END LOOP;
```

```
DBMS_OUTPUT.PUT_LINE('10 rows inserted.');
END;
/
```

20) Update Rows in Square Table

```
BEGIN
FOR i IN 1..10 LOOP
UPDATE square_table
SET square = i*i
WHERE num = i;
END LOOP;
DBMS_OUTPUT.PUT_LINE('Rows updated.');
END;
/
```

21) Delete Rows from Square Table

```
BEGIN
FOR i IN 1..10 LOOP
DELETE FROM square_table
WHERE num = i;
END LOOP;
DBMS_OUTPUT.PUT_LINE('Rows deleted.');
END;
/
```