**Exercise 1: Control Structures**

**Code:**

DECLARE

    v\_number        NUMBER := 15;

    v\_grade         CHAR(1) := 'B';

    v\_counter       NUMBER := 1;

    v\_limit         NUMBER := 5;

    v\_factorial\_num NUMBER := 4;

    v\_factorial\_res NUMBER := 1;

BEGIN

    -- 1. IF-THEN-ELSIF-ELSE Statement

    DBMS\_OUTPUT.PUT\_LINE('--- IF-THEN-ELSIF-ELSE Example ---');

    IF v\_number > 20 THEN

        DBMS\_OUTPUT.PUT\_LINE(v\_number || ' is greater than 20.');

    ELSIF v\_number >= 10 AND v\_number <= 20 THEN

        DBMS\_OUTPUT.PUT\_LINE(v\_number || ' is between 10 and 20 (inclusive).');

    ELSE

        DBMS\_OUTPUT.PUT\_LINE(v\_number || ' is less than 10.');

    END IF;

    DBMS\_OUTPUT.PUT\_LINE(CHR(10)); -- New line for better readability

    -- 2. CASE Statement

    DBMS\_OUTPUT.PUT\_LINE('--- CASE Statement Example ---');

    CASE v\_grade

        WHEN 'A' THEN

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

        WHEN 'B' THEN

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

        WHEN 'C' THEN

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

        ELSE

            DBMS\_OUTPUT.PUT\_LINE('Needs improvement.');

    END CASE;

    -- You can also use a searched CASE statement:

    CASE

        WHEN v\_grade = 'A' THEN

            DBMS\_OUTPUT.PUT\_LINE('Truly excellent.');

        WHEN v\_grade IN ('B', 'C') THEN

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

        ELSE

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

    END CASE;

    DBMS\_OUTPUT.PUT\_LINE(CHR(10));

    -- 3. Simple LOOP

    DBMS\_OUTPUT.PUT\_LINE('--- Simple LOOP Example ---');

    LOOP

        DBMS\_OUTPUT.PUT\_LINE('Simple Loop Iteration: ' || v\_counter);

        v\_counter := v\_counter + 1;

        EXIT WHEN v\_counter > v\_limit; -- Exit condition

    END LOOP;

    v\_counter := 1; -- Reset counter

    DBMS\_OUTPUT.PUT\_LINE(CHR(10));

    -- 4. WHILE LOOP

    DBMS\_OUTPUT.PUT\_LINE('--- WHILE LOOP Example ---');

    WHILE v\_counter <= v\_limit LOOP

        DBMS\_OUTPUT.PUT\_LINE('While Loop Iteration: ' || v\_counter);

        v\_counter := v\_counter + 1;

    END LOOP;

    v\_counter := 1; -- Reset counter

    DBMS\_OUTPUT.PUT\_LINE(CHR(10));

    -- 5. FOR LOOP (Numeric Range)

    DBMS\_OUTPUT.PUT\_LINE('--- FOR LOOP Example ---');

    FOR i IN 1..v\_limit LOOP

        DBMS\_OUTPUT.PUT\_LINE('For Loop Iteration: ' || i);

    END LOOP;

    DBMS\_OUTPUT.PUT\_LINE(CHR(10));

    -- FOR LOOP (Reverse Numeric Range)

    DBMS\_OUTPUT.PUT\_LINE('--- FOR LOOP (Reverse) Example ---');

    FOR i IN REVERSE 1..v\_limit LOOP

        DBMS\_OUTPUT.PUT\_LINE('For Loop (Reverse) Iteration: ' || i);

    END LOOP;

    DBMS\_OUTPUT.PUT\_LINE(CHR(10));

    -- Example: Calculate factorial using FOR LOOP

    DBMS\_OUTPUT.PUT\_LINE('--- Factorial Calculation with FOR LOOP ---');

    FOR i IN 1..v\_factorial\_num LOOP

        v\_factorial\_res := v\_factorial\_res \* i;

    END LOOP;

    DBMS\_OUTPUT.PUT\_LINE('Factorial of ' || v\_factorial\_num || ' is: ' || v\_factorial\_res);

    DBMS\_OUTPUT.PUT\_LINE(CHR(10));

    -- 6. GOTO Statement (Use sparingly, generally avoid in modern PL/SQL)

    DBMS\_OUTPUT.PUT\_LINE('--- GOTO Example ---');

    IF v\_number = 15 THEN

        GOTO jump\_label;

    END IF;

    DBMS\_OUTPUT.PUT\_LINE('This line will be skipped if v\_number is 15.');

    <<jump\_label>> -- Label for GOTO

    DBMS\_OUTPUT.PUT\_LINE('Reached GOTO label. This demonstrates jumping to a specific point.');

EXCEPTION

    WHEN OTHERS THEN

        DBMS\_OUTPUT.PUT\_LINE('An error occurred: ' || SQLERRM);

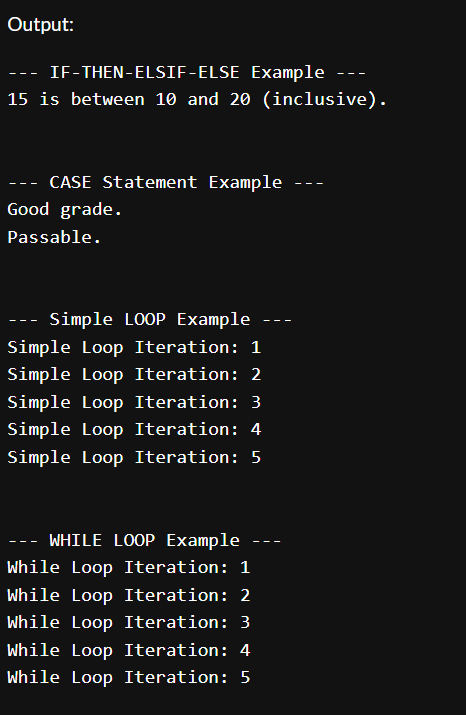
END;

/

-- To run this code in SQL\*Plus or SQL Developer, you need to enable server output:

-- SET SERVEROUTPUT ON;

**Output:**

****

