Case Statement

- The format of a case statement is
- **case *expression is***
  - **when *choices => sequential-statements***
  - **when *choices => sequential-statements*** [**when others => *sequential-statements]***
- **end case;**

Null Statement

The statement

null;

is a sequential statement that does not cause any action to take place and execution continues with the next statement. One example of this statement's use is in an if statement or in a case statement where for certain conditions, it may be useful or necessary to explicitly specify that no action needs to be performed.

Loop Statement

- ***A loop statement is used to iterate through a set of sequential statements. The syntax of a loop statement is***

- [*loop-label :] iteration-scheme* ***loop***
  - *sequential-statements*
- **end loop [** ***loop-label] ;***

Loop Statement cont..

- There are three types of iteration schemes. The first is the for iteration scheme that has the form

for *identifier in range*

An example of this iteration scheme is

```
FACTORIAL := 1;
for NUMBER in 2 to N loop
    FACTORIAL := FACTORIAL * NUMBER;
end loop;
```

Loop Statement cont..

- The second form of the iteration scheme is the while scheme that has the form

while *boolean-expression*

An example of the while iteration scheme is

J:=0;SUM:=10;

WH-LOOP: **while J < 20 loop - This loop has a label, WH_LOOP.**

SUM *:= SUM * 2;*

J:=J+3;

end loop;

Loop Statement cont..

- The third and final form of the iteration scheme is one where no iteration scheme is specified.
- In this form of **loop statement, all statements in the loop body are repeatedly executed until some other action causes it to exit the loop.**
- **These actions can be caused by an exit statement, a next statement, or a return statement.**

Loop Statement cont..

- Here is an example.

```
SUM:=1;J:=0;
    L2: loop
    J:=J+21;
    SUM := SUM* 10;
    exit when SUM > 100;
end loop L2;
```

Exit Statement

- The exit statement is a sequential statement that can be used only inside a loop. It causes execution to jump out of the innermost loop or the loop whose label is specified. The syntax for an exit statement is
- **exit [*loop-label] [when condition]:***
- If no loop label is specified, the innermost loop is exited.

Next Statement

- The next statement is also a sequential statement that can be used only inside a loop. The syntax is the same as that for the exit statement except that the keyword next replaces the keyword exit. Its syntax is
- **next [*loop-label] [when condition];***

- The **next statement results in skipping the remaining statements in the current iteration of the specified loop and execution resumes with the first statement in the next iteration of this loop. If no loop label is specified, the innermost loop is assumed.**

Next Statement cont..

- In contrast to the exit statement that causes the loop to be terminated (i.e., exits the specified loop), the next statement causes the current loop iteration of the specified loop to be prematurely terminated and execution resumes with the next iteration.

Next Statement cont..

Here is an example.

```
for J in 10 downto 5 loop
    if (SUM < TOTAL_SUM) then
        SUM := SUM +2;
    elsif (SUM = TOTAL_SUM) then
        next;
    else
        null;
    end if;
        K:=K+1;
end loop;
```