

If ..Then .. Else

- Normally, the instructions following the if statement cannot be executed if the condition is not true.
- An **if - then statement** followed by an optional **else statement**, executes when the boolean expression is false.



If Then Else Example

- Writeln('Who is the president of Zambia');
- Readln(ans);
- **If** (ans = 'Lungu') **Then**
- score := score + 1 { if this does not execute, }
- **Else** Writeln('sorry, you've got it wrong!'); { then this executes }



Nested If Statements

- A nested if statement is an if statement within another if statement, it is in the form:
- **If** (this happens) **then** {if 1}
- **If** (this happens) **then** {if 2}
- (do this) etc...
- **Else** (do this) {if 2}
- **Else** (do this) etc... {if 1}



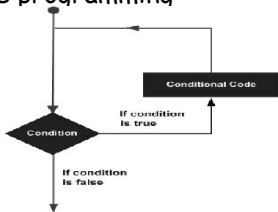
Nested If Example

- Writeln('Enter sex');
- Readln(ans);
- **If** (ans = 'Female') **Then**
- Writeln('don't come for class tomorrow')
- **Else**
- **If** (ans='Male') **Then**
- Writeln('come for class tomorrow')
- **else**
- Writeln('no comment');



Loops

- A loop statement allows us to execute a statement or group of statements multiple times.
- the general form of a loop statement in most of the programming languages is



Loops

- Pascal programming language provides the following types of loop constructs to handle looping requirements.
 - while-do loop
 - For - Loop
 - The Repeat-Until Loop
 - Nested Loops



while-do loop

- A **while-do** loop statement in Pascal allows repetitive computations till some test condition is satisfied.
- In other words, it repeatedly executes a target statement as long as a given condition is true.
- The syntax of a while-do loop is
 - while (condition) do S;



while-do loop

- Where, **condition** is a Boolean or relational expression whose value would be true or false and **S** is a simple statement or group of statements within BEGIN ... END block.



while-do loop example

- program whileLoop;
- var a: integer;
 - begin a := 10;
 - while a < 20 do
 - begin
 - writeln('value of a: ', a);
 - a := a + 1;
 - end;
 - end.



For - Loop

- Executes a sequence of statements multiple times.
- In other words, it repeatedly executes a target statement for predetermined or fixed times.
- The syntax for the for-do loop in Pascal is as follows:
 - for < variable-name > := < initial_value > to [down to] < final_value > do S;



For - Loop

- Where, the *variable-name* specifies a variable called control variable or index variable;
- *initial_value* and *final_value* values are values that the control variable can take; and
- S is the body of the for-do loop that could be a simple statement or a group of statements.



For - Loop example

- program ForLoop;
- var a: integer;
 - begin
 - For a := 10 to 20 do
 - begin
 - writeln('value of a: ', a);
 - end;
 - end.



The Repeat-Until Loop

- Unlike for and while loops, which test the loop condition at the top
- the **repeat ... until** loop in Pascal checks its condition at the bottom of the loop.
- A repeat ... until loop is similar to a while loop, except that a repeat ... until loop is guaranteed to execute at least one time.



The Repeat-Until Loop

- repeat
- S1;
- S2;
- ...
- ...
- Sn;
- until condition;



The Repeat-Until Loop Example

- program repeatUntilLoop;
- var a: integer;
- begin
 - a := 10;
 - (* repeat until loop execution *)
 - repeat
 - writeln('value of a: ', a);
 - a := a + 1
 - until a = 20; end.



While do and Repeat Until

- In While...do and Repeat...until there is a chance to get an infinite loop.
- Simply give an impossible conditions.
Example :
- While 2<4 do
- Begin
 - write('bla bla bla ...')
 - End
 (2 is always less than 4, so While...do get an infinite loop)



- OR
- Repeat
 - (bla bla bla ...)
 - Until 3>5;
 (3 is never be able to exceed 5, so Repeat...until goes forever)



Questions

