

Lesson 5: The CASE-OF Statement

Before continuing with this lesson, make sure that you have learnt to use 'if statement' (lesson 4)

In a nutshell, this lesson will cover:

- The Syntax of the CASE-OF Statement
- The CASE-OF-ELSE Statement

The Syntax of the CASE-OF Statement

So far, you have learned how to use an 'if statement'. The **Case-Of** statement is used to execute multiple value checks on the same variable. When one of them matches the value of the variable in question, the corresponding code executes. The If-Statement can be used to achieve the same function but in some cases the case-of statement is preferred to the if statement. This is because the latter keeps the code neat and compact while the if statement tends to end up with long pieces of code.

Here is how it works:

Case { variable of type: *integer / character / string* } **of**

 { value : code.. }

 { value : code.. }

...

End; {End Case}

When using String or Character data types for the Case-Of statement, the values need to be enclosed within single quotes. In the following program you will notice the difference between case statement over the if statement. The first program is written using the if statement:

```
Program Program1a_Lesson5;
Uses Crt;

Label Return; { used with a goto statement; not recommended }

Var
    SEL : Integer;
    YN : Char;

Begin
    Return: Clrscr;
```

```

Writeln('[1]. PLAY GAME');
WRITELN('[2]. LOAD GAME');
WRITELN('[3]. MULTIPLAYER');
WRITELN('[4]. EXIT GAME');
Writeln('note: Do not press anything except');
Writeln('numbers; otherwise an error occurs!');
Readln(SEL);

If SEL = 1 Then
Begin
    Writeln('You will soon be able to create');
    Writeln('games using Pascal Programming :-)');
    Delay(2000);
    Goto Return;
End;

If SEL = 2 Then
Begin
    Writeln('Ahhh... no saved games');
    Delay(2000);
    Goto Return;
End;

If SEL = 3 Then
Begin
    Writeln('networking or 2 players?');
    Delay(2000);
    Goto Return;
End;

If SEL = 4 Then
Begin
    Writeln('Are you sure you want to Exit?');
    YN := Readkey;
    If YN = 'y' Then
        Begin
            Writeln('Good Bye...');
            Delay(1000);
            Halt; {EXIT PROGRAM}
        End;

    If YN = 'n' Then
        Goto Return;
    End;
End.

```

Now, the folowing program is written using the case-of statement instead of the if statements and the outcome is perfectly the same.

```

Program Program1b_Lesson5;
Uses Crt;

Label Return; {use of the goto statement is not recommended.. avoid it}

Var
    SEL : Integer;
    YN : Char;

Begin
    Return:Clrscr;
    Writeln('[1].PLAY GAME');
    WRITELN('[2].LOAD GAME');
    WRITELN('[3].MULTIPLAYER');
    WRITELN('[4].EXIT GAME');
    Writeln('note: Do not press anything except');
    Writeln('numbers; otherwise an error occurs!');
    Readln(SEL);

    Case SEL of
        1 : Begin
            Writeln('You will soon be able to create');

```

```

                                Writeln('games using Pascal Programming :-)');
                                Delay(2000);
                                Goto Return;
                                End;
                                2 : Begin
                                Writeln('Ahhh... no saved games');
                                Delay(2000);
                                Goto Return;
                                End;
                                3 : Begin
                                Writeln('networking or 2 players?');
                                Delay(2000);
                                Goto Return;
                                End;
                                4 : Begin
                                Writeln('Exit?');
                                YN := Readkey;
                                Case YN of {a sort of a nested case statement}
                                    'y' : Begin
                                                Writeln('Good Bye...');
                                                Delay(1000);
                                                Halt;
                                                End;
                                    'n' : Goto Return;
                                End; {End Case 2}
                                End; {Close Conditional Expression 4}
                                End; {End Case 1}
End.

```

The CASE-OF-ELSE Statement

Again this is similar to the **if..then..else** statement. Study the program below to learn how to use the 'else' term following the 'case statement':

```

Program Program2_Lesson5;
Uses Crt;
Label Return; { avoid it }
Var YN : Char;

Begin
    Return: ClrScr;
    Writeln('Exiting?');
    YN := Readkey;
    Case YN of
        'y' : Halt;
        'n' : Begin
                    Writeln('What are you going to do here, anyway?');
                    Delay(2000);
                    Halt;
                    End;
        Else
        Begin
            Writeln('Press either 'y' for yes');
            Writeln('or 'n' for no.. please try again..');
            Delay(3500);
            ClrScr;
            Goto Return;
        End;
    End; {CASE}
End. {PROGRAM}

```

Basically, what the code does above is that if the input is neither 'y' nor 'n', then execution flow falls to the 'else' of the case statement - its like an 'if all else fails' and works like the last 'else' of the if statement.

