PASCAL - VARIABLE SCOPE

A scope in any programming is a region of the program where a defined variable can have its existence and beyond that variable cannot be accessed. There are three places, where variables can be declared in Pascal programming language −

- Inside a subprogram or a block which is called local variables
- Outside of all subprograms which is called global variables
- In the definition of subprogram parameters which is called formal parameters

Let us explain what are **local** and **global** variables and formal parameters.

Local Variables

Variables that are declared inside a subprogram or block are called local variables. They can be used only by statements that are inside that subprogram or block of code. Local variables are not known to subprograms outside their own. Following is the example using local variables. Here, all the variables *a*, *b* and *c* are local to program named *exLocal*.

```
program exLocal;
var
   a, b, c: integer;

begin
   (* actual initialization *)
   a := 10;
   b := 20;
   c := a + b;

   writeln('value of a = ', a , ' b =  ',  b, ' and c = ', c);
end.
```

When the above code is compiled and executed, it produces the following result −

```
value of a = 10 b = 20 c = 30
```

Now, let us extend the program little more, let us create a procedure named display, which will have its own set of variables *a*, *b* and *c* and display their values, right from the program *exLocal*.

```
program exLocal;
var
   a, b, c: integer;
procedure display;

var
   a, b, c: integer;
begin
   (* local variables *)
   a := 10;
   b := 20;
   c := a + b;

   writeln('Winthin the procedure display');
   writeln('value of a = ', a , ' b =  ',  b, ' and c = ', c);
end;

begin
   a:= 100;
   b:= 200;
   c:= a + b;
```

```
   writeln('Winthin the program exlocal');
   writeln('value of a = ', a , ' b =  ',  b, ' and c = ', c);
   display();
end.
```

When the above code is compiled and executed, it produces the following result –

```
Within the program exlocal
value of a = 100 b = 200 c = 300
Within the procedure display
value of a = 10 b = 20 c = 30
```

Global Variables

Global variables are defined outside of a function, usually on top of the program. The global variables will hold their value throughout the lifetime of your program and they can be accessed inside any of the functions defined for the program.

A **global** variable can be accessed by any function. That is, a global variable is available for use throughout your entire program after its declaration. Following is an example using **global** and **local** variables –

```
program exGlobal;
var
   a, b, c: integer;
procedure display;
var
   x, y, z: integer;

begin
   (* local variables *)
   x := 10;
   y := 20;
   z := x + y;

   (*global variables *)
   a := 30;
   b:= 40;
   c:= a + b;

   writeln('Winthin the procedure display');
   writeln(' Displaying the global variables a, b, and c');

   writeln('value of a = ', a , ' b =  ',  b, ' and c = ', c);
   writeln('Displaying the local variables x, y, and z');

   writeln('value of x = ', x , ' y =  ',  y, ' and z = ', z);
end;

begin
   a:= 100;
   b:= 200;
   c:= 300;

   writeln('Winthin the program exlocal');
   writeln('value of a = ', a , ' b =  ',  b, ' and c = ', c);

   display();
end.
```

When the above code is compiled and executed, it produces the following result –

```
Within the program exlocal
value of a = 100 b = 200 c = 300
Within the procedure display
Displaying the global variables a, b, and c
value of a = 30 b = 40 c = 70
```

```
Displaying the local variables x, y, and z
value of x = 10 y = 20 z = 30
```

Please note that the procedure display has access to the variables a, b and c, which are global variables with respect to display as well as its own local variables. A program can have same name for local and global variables but value of local variable inside a function will take preference.

Let us change the previous example a little, now the local variables for the procedure display has same names as *a*, *b*, *c* −

```
program exGlobal;
var
   a, b, c: integer;
procedure display;

var
   a, b, c: integer;

begin
   (* local variables *)
   a := 10;
   b := 20;
   c := a + b;

   writeln('Winthin the procedure display');
   writeln(' Displaying the global variables a, b, and c');

   writeln('value of a = ', a , ' b =  ',  b, ' and c = ', c);
   writeln('Displaying the local variables a, b, and c');

   writeln('value of a = ', a , ' b =  ',  b, ' and c = ', c);
end;

begin
   a:= 100;
   b:= 200;
   c:= 300;

   writeln('Winthin the program exlocal');
   writeln('value of a = ', a , ' b =  ',  b, ' and c = ', c);

   display();
end.
```

When the above code is compiled and executed, it produces the following result −

```
Within the program exlocal
value of a = 100 b = 200 c = 300
Within the procedure display
Displaying the global variables a, b, and c
value of a = 10 b = 20 c = 30
Displaying the local variables a, b, and c
value of a = 10 b = 20 c = 30
```