

PASCAL - SUBPROGRAM CALL BY VALUE

http://www.tutorialspoint.com/pascal/pascal_call_by_value.htm

Copyright © tutorialspoint.com

The **call by value** method of passing arguments to a subprogram copies the actual value of an argument into the formal parameter of the subprogram. In this case, changes made to the parameter inside the function have no effect on the argument.

By default, Pascal uses **call by value** method to pass arguments. In general, this means that code within a subprogram cannot alter the arguments used to call the subprogram. Consider the procedure *swap* definition as follows.

```
procedure swap(x, y: integer);
var
    temp: integer;

begin
    temp := x;
    x := y;
    y := temp;
end;
```

Now, let us call the procedure *swap* by passing actual values as in the following example –

```
program exCallbyValue;
var
    a, b : integer;
(*procedure definition *)
procedure swap(x, y: integer);

var
    temp: integer;

begin
    temp := x;
    x := y;
    y := temp;
end;

begin
    a := 100;
    b := 200;
    writeln('Before swap, value of a : ', a );
    writeln('Before swap, value of b : ', b );

    (* calling the procedure swap by value *)
    swap(a, b);
    writeln('After swap, value of a : ', a );
    writeln('After swap, value of b : ', b );
end.
```

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

```
Before swap, value of a :100
Before swap, value of b :200
After swap, value of a :100
After swap, value of b :200
```

Loading [MathJax]/jax/output/HTML-CSS/fonts/TeX/fontdata.js