

Forum:[Hack Forums](#)

Thread: [Pascal Programming Tutorial - 1](#)

Deadbyte - February 21, 2015, 6:55 pm

Pascal Programming Tutorial - 1

Intro

I presume this is the right section for this tutorial. Since Delphi is Object Pascal. Pascal is a programming language. It was created in 1970 by Niklaus Wirth, to help people learn how to make computer programs.

Now, there are many different dialects of the language, so a Pascal program written for one compiler can not easily be used with another one.

This tutorial has multiple sources myself and other official Pascal websites.

[img]http://www-history.mcs.st-and.ac.uk/BigPictures/Pascal_2.jpeg[/img]

Blaise Pascal (19 June 1623 – 19 August 1662)

Turbo Pascal is a software development system that includes a compiler and an integrated development environment (IDE) for the Pascal programming language. From version 6 both a lower-priced Turbo Pascal and more expensive Borland Pascal were produced; Borland Pascal had more libraries and standard library source code. The name Borland Pascal is also used more generically for Borland's dialect of Pascal.

Preview:

[img]<http://upload.wikimedia.org/wikipedia/commons/f/f5/LagrangePAS.jpg>[/img]

Programming Basics Lesson 1

In a program, you must always obey the rules of the language, in our case, the Pascal language. A natural language has its own grammar rules, spelling and sentence construction. The Pascal programming language is a high level language that has its own syntax rules and grammar rules. As you go along with the lessons, you must note what you can do and what you cannot do in writing a Pascal program. A very simple program is shown below:

PHP Code

```
<?php
```

```
Program Lesson1_Program1;
```

```
Begin
```

```
Write('Hello World. Prepare to learn PASCAL!!');
```

```
Readln;
```

```
End.
```

The program is written only to display the message : 'Hello World. Prepare to learn PASCAL!!' - an introductory message that is displayed to you whenever you are going to learn a new programming language. This is simply shown on the screen. So, to display any message on the screen, you should use 'write' (or 'writeln'). The 'readln' statement, here is used as to 'stop' the program and wait until the user presses enter. If the 'readln' statement is missing in this program, then the message is displayed on the screen without giving any chance for the user to read it and obviously halts! Try running this program with and without the 'readln' statement and notice the difference. I suggest you see it!! Now, look at this:

PHP Code

```
<?php
```

```
Program Lesson1_Program2;begin
```

```
Write('Hello World. Prepare to learn PASCAL!!');Readln;End.
```

This program also runs perfectly as the previous one. The only difference is: neatness and friendliness.

This first program is, what is commonly referred to in programming, as 'indented'. Indentation is a must in writing programs as it aids in the way the code is written ie. neater. Indentation also helps with debugging and code presentation. You will note how I indent programs.

String Variables Lesson 2

Soon, you should learn how to input text by the user using 'string variables'. The following program is written showing an example of a string variable, prompting the user to input his name, whatsoever:

PHP Code

```
<?php
```

```
Program Lesson2_Program1;
```

```
Var name, surname: String;
```

```
Begin
```

```
Write('Enter your name:');
```

```
readln(name);
```

```
Write('Enter your surname:');
```

```
readln(surname);
```

```
writeln;{new line}
```

```
writeln;{new line}
```

```
Writeln('Your full name is: ',name,' ',surname);
```

```
Readln;
```

```
End.
```

If we take a look at this program, we notice a new variable type : 'String'. Both the name and surname variables are of type string. When the program is run and prompts the user to input his name, the name which is keyed in by the user goes directly to its place in the memory called 'name'. Same occurs to surname. *I'd like to remind you that the variables 'name' and 'surname' are not reserved words, but are used by the programmer as his variables. I could have used 'n' instead of 'name' and similarly 'sname' instead of 'surname'. The two 'idle' writeln's in lines 9 and 10 are used to move on for a new line. In this case, 2 lines are cleared. The next message displays the full name of the user using the above format. If a string variable is required to be displayed on screen, it should be put in between inverted commas and commas if it is concatenated with another message; example:

Constants and the Assignment Statement Lesson 3

Apart from variables, there are also items in the program which are referred to as 'constants'. Unlike variables, constants keep their value or string unchanged for the whole program. Here I have made a program, not so much different from the previous one:

PHP Code

```
<?php
```

```
Program Lesson2_Program2;
```

```
Va
```

```
    surname: String;
```

```
Const {the reserved word 'const'
```

```
    is used to initialize constants}
```

```
    name = 'Victor';
```

```

Begin
Write('Enter your surname:');
readln(surname);
writeln;
writeln;
Writeln('Your full name is: ',name,' ',surname);
Readln;
End.

```

In the above program, the constant 'name' is assigned to as 'Victor' and is of type string. However, in other cases, you might have used integer constants (whole numbers), i.e.:

```

Const
age = 15;
The constant 'age' is a value that could be used wherever it is required in a program. Example:
age2 := 15;
age2 := age + 15;

```

The above example shows an addition of the value of the variable 'age' with the value 15. The value of the constant 'age' remains 15, but the value of the variable 'age2' becomes 30. The assignment statement is not only used for additions, but is also used to assign a variable: text if it is a string variable or a numeric value if it is an integer variable.

Special Reserved Words of the CRT Unit: ClrScr(), GotoXy(), etc... Lesson 4

Before I teach you the if statements and for loops, I would like to give you an idea of some pascal functions which are quite useful. This would require your program to include the library 'crt.tpu'. To include a library in the program, one should use the reserved word 'uses', because it is used to call a library of functions and procedures. Here is the program of lesson 2 (program 3) which is better handled and more user-friendly:

PHP Code

```
<?php
```

```
Program lesson3_Program1;
```

```
Uses Crt; {We will make use of the crt library}
```

```

Var PD, Dname, Cmodel : String;
    CostPD, TCostPD, Distance : Real;
    {real is a decimal (described later)}

```

```

Begin
textbackground(brown); {background colour}
ClrScr; {Clear screen with a brown colour.
    Try run the program without this!!!}
TextColor(lightgreen); {text colour}
TCostPD := 0;
Writeln('This program prompts you to '+'
    +'input the cost per litre of');
Writeln('the petrol/diesel you spend in and '+'
    +'the average distance you travel');
Writeln('with your car every week. Then, '+'
    +'the computer calculates the total cost');
Writeln('you spend in fuel every week. ');
Readkey; {program move on as soon as a key is pressed}
ClrScr;{short for clear screen}
GotoXy(28,3);
{^move to a position on the screen:

```

```

    x (horizontal), y (vertical)}
Write('Diesel or Petrol? Type p or d: ');
PD := Readkey;
{^as soon as a key is pressed,
 it is stored in the variable 'PD'}
GotoXy(30,4);
Write('Name Of Driver: ');
Readln(Dname);
GotoXy(30,5);
Write('Car Model: ');
Readln(Cmodel);
GotoXy(29,6);
Write('Cost of Diesel/Petrol: (£) ');
Readln(CostPD);
GotoXy(8,7);
Writeln('Average distance you travel with '+
    +'your car every week: (kilometres) ');
Readln(Distance);
ClrScr;
GotoXy(28,3);
Writeln('Name of Driver:',Dname);
GotoXy(31,4); Delay(500);
Writeln('Car Model:',Cmodel);
GotoXy(32,5); Delay(500);
Writeln('Diesel/Petrol:',PD);
GotoXy(8,6); Delay(500);
Writeln('Average distance covered '+
    +'every week: ',Distance:1:2,'Km');
GotoXy(25,7); Delay(500);
Writeln('Cost of ',PD,' per litre: £',CostPD:1:2,'/litre');
Writeln; Delay(500);
Writeln;
TCostPD := Distance * CostPD;
GotoXy(21,10);
Writeln('Total cost of ',PD,' per week:£',TCostPD:1:2);
TCostPD := 0;
GotoXy(21,12);
Writeln('Total cost of ',PD,' per week:'+
    +'£',(Distance * CostPD):1:2);
GotoXy(18,14);
Writeln('Total cost of ',PD,' per week:£',Distance * CostPD);
readln;
End.

```

The 'Crt' (short for cathode-ray tube) library has a wide range of functions and procedures that you will use so frequently. Some of them are listed in the table below. There are many similar libraries, such as 'Strings' (you will be learning something on this later) and 'Dos'.

Program Control Lesson 5

Now, it is time to learn the most important rules of programming: the if statements - decision making, for loops and the repeat-until loop. Almost, these 3 general programming constructs are common in every programming language and you have to make sure that when you have finished reading this lesson, make sure that you have practiced them enough before continuing with learning pascal because they are of outmost importance. If you fall in doubt about these programming

constructs, then try to use the forums, describing your problem.

The 'if statement' executes a the proceeding statement(s) conditionally. This means that if an action comes to be true, then the statement(s) proceeding the if statement are executed, else these statements are skipped. It works like this:

If this happens(action), then do this(reaction, if action is true).

OR:

If this happens(action), then do this(reaction, if action is true), else do this(reaction, if action is false).

In Pascal, the 'if statement' should be written as follows:

If conditional expression then code ... ;{if one action}

OR:

If conditional expression then Begin instructions ... End; {if more than one action is required}

Note that you should not use an assignment statement in the 'if' construct, otherwise the compiler will raise a syntax error. I.e.:

Wrong:

If x := 20 then x := x + 1; {the underlined character must be excluded}

Correct:

If x = 20 then x := x + 1; {only an equal sign is used for comparison}

A program is shown below as an example of how the 'if statement' works:

PHP Code

```
<?php
```

```
Program lesson4_Program1;
```

```
Uses Crt;
```

```
Label 1; {this is used with a goto statement}
```

```
Var Sel: String;
```

```
    N1,N2, Total : Real;
```

```
    YN : Char; {this is a character variable type,  
               which holds single characters ONLY}
```

```
Begin
```

```
1:Clrscr;
```

```
Total := 0; {always initialise integer/real variables}
```

```
GotoXy(4,3);
```

```
Writeln('1.Addition');
```

```
GotoXy(4,4);
```

```
Writeln('2.Subtraction');
```

```
GotoXy(4,5);
```

```
Writeln('3.Exit');
```

```
GotoXy(6,8);
```

```
Write('Select: ');
```

```
Sel := Readkey;
```

```
If Sel = '1' {action} then
```

```
    Begin {more than one statement}
```

```
        ClrScr;
```

```
        Write('Input No.1:');
```

```
        Readln(N1);
```

```
        Write('Input No.2:');
```

```
        Readln(N2);
```

```
        Total := N1 + N2;
```

```
        Writeln('Addition: ',N1:2:3,' + ',N2:2:3,' = ',Total:2:3);
```

```
        Write('Press any key to continue...');
```

```
        Readkey;
```

```
        Goto 1;{this leads back to the beginning of the program,  
              otherwise the program terminates}
```

```

End; {Closing the if statement(begin)}
If Sel = '2' then
{note that the assignment statement
 is not used within an if statement}
Begin
  ClrScr;
  Write('Input No.1:');
  Readln(N1);
  Write('Input No.2:');
  Readln(N2);
  Total := N1 - N2;
  Write('Subtraction: ');
  Write(N1:2:3, ' - ', N2:2:3, ' = ', Total:2:3);
  Write('Press any key to continue...');
  Readkey;
  Goto 1;
End; {Closing the if statement}
If Sel = '3' then
Begin
  ClrScr;
  Write('Are you sure?(Y/N)');
  YN := Readkey;
  If YN = 'y' then Halt; {1 action, so no need of Begin..End}
  If YN = 'n' then Goto 1; {the goto statement is not
    recommended for excessive use}
End;
End.

```

In the above program, the 'goto' statement is used. So far, it has been a real damage to programs and it has produced unwanted confusions. I strongly suggest you not to use it repeatedly.

If..Then..Else

In a normal if statement, the 'reaction' cannot be performed if the condition is not true. But in an if..then..else statement, there is at least one set of statements to be performed. Let's take a look at the example below:

```

writeln('Who has discovered the land of America?');
Readln(ans);
If (ans = 'Christopher Colombus') then
  score := score + 1    {if this is false,}
ELSE
  writeln('sorry, you've got it wrong!');    {then this is true}

```

Note that if the 'else' term is included with an if statement, then there should be no semi-colon before the 'else' term; just as seen in the above example.

Pascal Programming Tutorial - 2 is coming soon.

aldypalypso - February 23, 2015, 8:46 am

Your tutorial is nice, but it should've been bigger in my opinion. Too small.

Deadbyte - February 23, 2015, 8:48 am

(Feb 23, 2015 09:46 AM)aldypalypso Wrote: [\[CITE\]](#) Your tutorial is nice, but it should've been bigger in my opinion. Too small.

I'll make part 2 this week I think.

borisganji1 - February 23, 2015, 8:55 am

You should really write the pros and cons about Pascal.

I know C++ and I don't see any upside learning this, C++ can probably do the same, while have a better documentation - but then again I know nothing about pascal :)

Regards

Deadbyte - February 23, 2015, 9:33 am

(Feb 23, 2015 09:55 AM)borisganji1 Wrote: [\[CITE\]](#) You should really write the pros and cons about Pascal.

I know C++ and I don't see any upside learning this, C++ can probably do the same, while have a better documentation - but then again I know nothing about pascal :)

Regards

I made the thread for educational purpose only. Pascal is useless now in my opinion, but a good first language for a beginner who has difficulties learning programming.

Recursive™ - February 23, 2015, 1:34 pm

I find Pascal a language perfectly appropriate for coding learners, allowing users to properly learn programming paradigms and fully experiment with a small extent of the capabilities of third generation languages. This language has really helped me properly understand C++ and Java. The guide could indeed be added to, although on the whole the guide is informative, easily read and laboriously thought through. Good job. I would however recommend not using the free pascal GUI because let's face it, that GUI is disgusting and horrible to use. Possibly encourage using Lazarus as much more user friendly.

Deadbyte - February 23, 2015, 2:06 pm

(Feb 23, 2015 02:34 PM)Forumnator Wrote: [\[CITE\]](#) I find Pascal a language perfectly appropriate for coding learners, allowing users to properly learn programming paradigms and fully experiment with a small extent of the capabilities of third generation languages. This language has really helped me properly understand C++ and Java. The guide could indeed be added to, although on the whole the guide is informative, easily read and laboriously thought through. Good job. I would however recommend not using the free pascal GUI because let's face it, that GUI is disgusting and horrible to use. Possibly encourage using Lazarus as much more user friendly.

I agree with everything you said. ?

Cyber_Hox - February 23, 2015, 3:49 pm

Good tutorial (moj prijateljku kada ce sledeci? :D)

Deadbyte - February 23, 2015, 3:53 pm

(Feb 23, 2015 04:49 PM)Cyber_Hox Wrote: [\[CITE\]](#) Good tutorial (moj prijateljku kada ce sledeci? :D)
This week I hope I will be free to do it ?