

Get with the program

These days, we generally rely on prepackaged software's built-in features to perform specific tasks. But, as Julian Moss explains in this beginners' guide, programming provides a hands-on approach to telling your PC who's boss

Today's PCs have more power than a supercomputer of the 1960s, are a thousandth of the cost and sit happily on the corner of a desk. Modern operating systems and applications hide any complexities, allowing us to instruct a PC simply by selecting items from a menu.

Though there's no longer any need to have programming knowledge, having even the most basic skills will enable you to automate tasks and customise your PC in ways Windows doesn't cater for. With a bit more knowledge you'll be able to write simple applications to perform tasks such as converting data across formats or maintaining a database. You could even create your own games.

Writing a program can be an absorbing and intellectual challenge – indeed, to many people it's a fun hobby. All you need is your PC and some spare time.

Under review

Windows Scripting Host

• www.microsoft.com; price: n/a
If you're interested in programming then there's one tool you can't ignore as it's already installed on your computer. The Windows Script Host, complete with interpreters for VBScript and JScript (a Java-like language) has shipped with every system since Windows 98.

You can write scripts to perform a variety of tasks, ranging from system management to the automation of your Office applications. Simply type code into a text editor, save it and run it. Unfortunately, this isn't as simple as it sounds. Without development aids such as syntax highlighting (which shows the elements of a program statement in different colours), context-sensitive help and an interactive debugger, getting your script working can be a slow process.

It's possible to use Microsoft FrontPage's Script Editor (see *Use*

Microsoft Script Editor to develop VBScripts on page 104), but this is designed for testing scripts embedded in web pages and insists on running everything in a web browser.

Koan Software's useful freeware tool VBS Editor (www.koansoftware.com/eng_index.htm), which is on this month's cover disc, is another option but it's certainly no substitute for a proper development environment.

It's worth overcoming the difficulties of scrip writing, though, because it's one of the most useful programming skills you can master. If you fancy giving it a go then visit the Microsoft Windows Scripting Home Page (<http://msdn.microsoft.com/scripting>). Here, you can download the

latest version of the software and essential accompanying documentation.

Other useful online resources include the Scripting Guide for Windows (www.winguides.com/scripting), Win32 Scripting (<http://cwashington.netreach.net>) and Born's Windows Scripting Host FAQ Page (http://ourworld.compuserve.com/homepages/guenter_born/wshbazaar/wsh2.htm). You'll also find *Microsoft Windows Script Host 2.0 Developer's Guide* useful (£28 from www.amazon.co.uk).

Borland Delphi

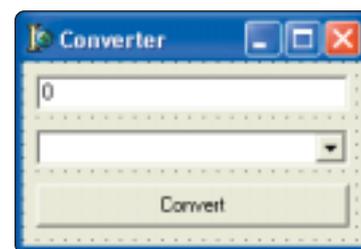
• www.borland.co.uk; from £79
Modelled on the Visual Basic tool, Borland Delphi follows a standard pattern for visual programming tools. Start by designing the

Your first Delphi program: designing the window

To demonstrate the principles of visual programming we'll create a simple conversion utility using Borland Delphi (www.borland.co.uk/delphi).



1 After starting Delphi, select File, New, Application. From the Standard tab of the Component palette, click on an Edit control then select your new application's window, Form1. With the mouse, drag the edit control so that its top and left co-ordinates both have the value eight and a width of about 180. Using the Object Inspector, change the Text property to '0'



2 Add, position and resize a ComboBox and a Button in the same manner. Using the Object Inspector, change the ComboBox's Style to 'csDropDownList'. Click the '...' button in the Items property then type a list of conversions to choose from such as 'Inches to Centimetres' and 'Centimetres to Inches'. For the Button, change its Name property to Convert. Select the form, change its name to Converter and resize it with the mouse so it fits round the three controls

```
OOO

#include <stdio.h>
#include <conio.h>
#define MAX 20

main()
{
static int arr[MAX];
int n, temp, pos;
int i, j, k;

printf("Enter the number of elements : ");
scanf("%d", &n);

printf("Enter the elements now :\n");
fflush(stdin);
for (i = 0; i < n; i++)
scanf("%d", &arr[i]);

i = 0;
pos = 0;
for (i = 0; i < n; i = (pos + 1))
{
j = pos + 1;

/* find out the jth element's appropriate position */
for (k = 0; arr[j] > arr[k] && k < j; k++);

/* shift the list elements till the element's position is
* found */
while (k < j)
{
temp = arr[j];
pos = j;
printf("\nk = %d\n", k);
arr[j] = arr[j - 1];
arr[k] = temp;

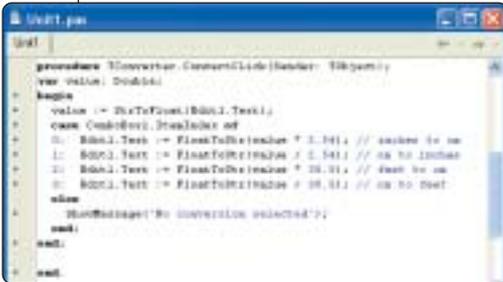
printf("\tIntermediate o/p is : ");
for (i = 0; i < n; i++)
printf("%2d ", arr[i]);
}

printf("\n\nInsertion sorted list is : \n");
for (i = 0; i < n; i++)
printf("%d ", arr[i]);

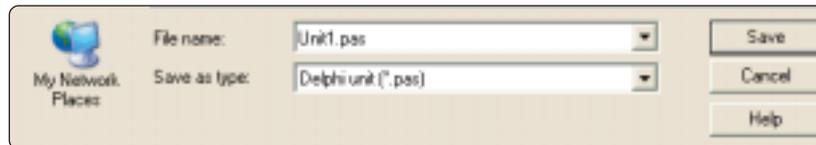
getch();
}
```

Your first Delphi program: adding the code

Now we must write the code that will run when the button is clicked. This will convert the number typed in the Edit box, depending on the option that is selected from the drop-down list.



1 Double-click the button and the code editor will open showing a procedure called TConverter.ConvertClick. Add the code so that the finished procedure looks like ours. The line starting '0:' will be used if the first item in the ComboBox list is selected. The formula for '1:' will be used for the second item and so on



2 Save your project by clicking the Save All button. Select a directory for the project using the buttons beside the Save In: field. Save the main program file as unit1.pas. You'll be prompted to save the project file – change the name to Convert because the EXE file that Delphi creates will be named after this

user interface, placing menus, buttons, input fields and so forth on a form. As each element is added, the relevant code is automatically inserted into the source file. This saves an enormous amount of work compared with writing Windows programs in a non-visual language like C.

Written code will be executed when a menu item is selected or a button pressed. Once that's done, you're ready to test it. Compilation is quick and the result is an EXE file. If there are errors, Delphi's debugger makes finding them a cinch. You can step through the program a line at a time and monitor the values of variables as you go.

Delphi is based on Pascal, a good language choice for the novice programmer as its syntax allows the compiler to spot errors that other languages might miss. But Delphi is a professional package capable of developing systems for large enterprises and, as such, it's expensive. The free version must only be used for training and even the Personal edition shouldn't be used for commercial work.

The University of Manchester Institute of Science and Technology has a Computing with Delphi tutorial (www.me.

umist.ac.uk/delphi.htm). The tutorials at DelphiLand (www.festra.com/eng/index.html) are also worth a look.

LCC-Win32

- www.cs.virginia.edu/~lcc-win32/: price: free

This free C compiler and development system runs under Windows. It's ideal for the novice programmer but is still a high-quality tool and has been used to develop add-ins for the shoot-'em-up title, Quake.

The development environment, called Wedit, lacks frills such as toolbars but has everything you need. You can create both command-line and Windows programs. The former tool is easier for the beginner as you can use the standard functions 'scanf' and 'printf' for input and output.

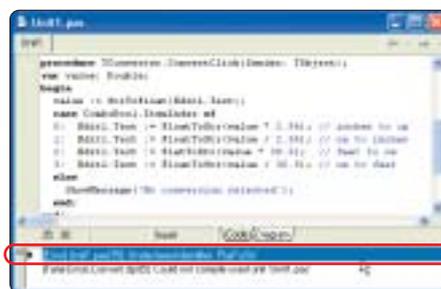
Wedit has a good debugger, too, which lets you set break points, step through your code and inspect the values of variables if the program doesn't work.

If you want to try your hand at Windows programming then Wedit has a wizard that creates an initial skeleton program as well as an editor that enables you to visually design your program window or dialog box. But LCC-Win32 is not a visual programming tool and plenty more work is needed to make a Windows interface function as it should.

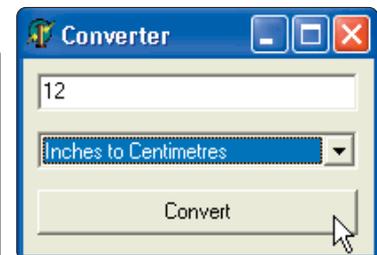
If you'd like to try using C you'll find a great introduction to programming at

Your first Delphi program: compiling and running

To compile and run your program, you need to click the Run button (the green arrowhead). Alternatively, press F9.



1 If there are no mistakes in your code, Delphi will turn it into a compiled executable file and run it. If there is an error, the compiler will display an error message similar to the one highlighted. In that case, you'll have to work out what you did wrong, correct it and try again



2 If the program is compiled correctly Delphi will run it. Test it by typing a number into the input Edit field, selecting a conversion and clicking Convert. Note how Delphi's built-in error checking displays a suitable message if you type an invalid figure into the input box. Once the program is finished you can run it whenever you want, without starting Delphi, using a shortcut to the EXE file in the project directory

Use Microsoft Script Editor to develop VBScripts

If you have Office 2000 or later you can use Microsoft FrontPage's Script Editor to edit and debug scripts. You'll get the benefits of colour syntax highlighting, help prompts and a debugger that lets you set break points and step through your script line by line. If you have the Script Editor installed on your PC, you'll find it in the folder `c:\Program Files\Microsoft Visual Studio\Common\IDE\IDE98`. The filename is `mse.exe`. However, there are a couple of things you need to do before you can use this editor to create VBScripts.

First, you need to enable support for standalone script files, since the Microsoft Script Editor was intended only for editing scripts in web pages. This entails adding a couple of values to the Registry. To make this easier, we've included a file called `mse.reg` on this month's cover disc. Just double-click the file to make the changes.

Next, you must make a file called new windows script file.wsf which contains the following:

```
<job>
<SCRIPT LANGUAGE="VBScript">
</SCRIPT>
</job>
```

Now save it in the folder `c:\Program Files\Microsoft Visual Studio\Common\IDE\IDE98\NewFileItems`. To save you some work we've created this for you as well, so you can just drag and drop it from the CD.

Start the Script Editor and select Tools, Options. Expand the HTML option and click on the HTML Editor. Under Initial View, Start HTML Pages and select Source. Click ok to save the changes.

You can now use Script Editor to edit and test VBScripts. To do this, select File, New File, New Windows Script File. Type your script code between the tags `<SCRIPT ...>` and `</SCRIPT>` in the new file. You must also force the file type extension to `WSF` when you save the file because the Script Editor defaults to `HTM`. An unavoidable annoyance is that a browser window starts when you run or debug your script from within the editor. You may also see a warning about ActiveX controls on the page, but you can safely ignore this.

www.howstuffworks.com/c.htm/printable. You should also visit Q Software Solutions (www.q-software-solutions.com) which has a short introductory tutorial on using LCC-Win32. Finally, there's some specific information at www.geocities.com/siliconvalley/station/1177/tutors.html.

DarkBasic Professional

- www.darkbasicpro.com; \$99

DarkBasic is designed specifically for creating 3D games, screensavers and presentations. Because of this, it requires some reasonably powerful hardware with 3D-accelerated graphics. Created programs run in full screen mode under DirectX.

DarkBasic's programming language is closer to traditional Basic than Microsoft's versions so it's fairly easy to learn. The downloadable demo lacks tutorials or

examples, though. These are only provided in the full product. The package includes all the tools required to create working programs: a code editor, compiler, debugger and context-sensitive online help for every function and command.

With commands such as 'make object', 'position object' and 'rotate object' you'll be creating programs with animated 3D graphics in no time at all. But there's plenty of power to keep you challenged and produce professional-looking games. There's support for textures and, for the advanced game programmer, textures and objects can be imported from external designers including a Quake editor.

Newcomers should log on to www.darkbasic.net. You'll find tutorials, code snippets and online forums where you can chat with other users.

History file

Back at the dawn of data processing, when a computer cost the earth and needed an air-conditioned room the size of a supermarket to house it, getting a system to work meant telling it exactly what to do, one step at a time. What's worse, you had to tell the computer in its own language: machine code. If you couldn't speak the lingo then you handed over the job to a programmer.

Digital dawn

The development of the digital computer began in 1945 after the mathematician John von Neumann came up with the concept of the 'stored program computer'. Instead of complex task-specific hardware that needed to be physically rewired for each job, von Neumann proposed that computer processors should be simple but able to read and store complex sets of instructions called programs.

An important feature of von Neumann's proposal was the conditional control transfer which allowed the result of one instruction to determine which one was performed next. Another of his ideas was the library: a collection of instructions for performing specific tasks that could be used by other programs whenever needed.

In 1949 the first programming language, Short Code, was designed. It suffered from one major disadvantage: the programmer had to convert it by hand into binary machine code (ones and zeroes).

Two years later programmer Grace Hopper, working for the US Navy, tired of this tedious task and wrote the first application that would convert programming language into machine code automatically. Called a compiler, this tool speeded up program development no end.

Modern languages

In 1957 IBM designed Fortran, a language that coded complex calculations but couldn't process data or text. Developed in 1960, Cobol was the first programming language developed for business use. Features included records, which were made up of different types of information such as name, address and date of birth.

Because the storage capacity of computers was limited (even in the early 1970s a system might have as little as

The quizmaster

For our VBScript example we've chosen a simple quiz, which you can easily modify to create a fun game or learning aid for the kids.

```

Title = "Quiz"
RightAnswers = 0

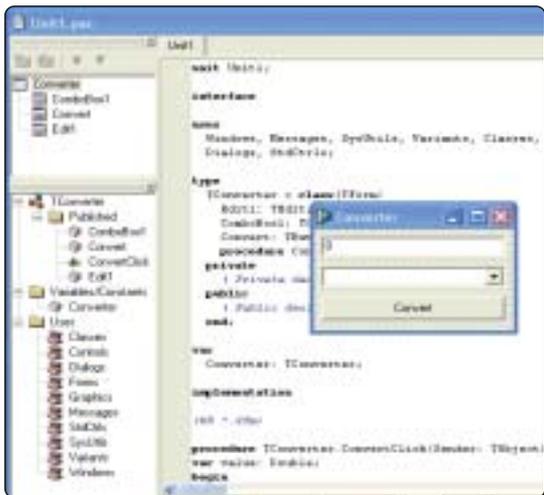
Ask "Have a small robot which says like to eat?", "Bacon"
Ask "Have a yellow fruit that says like to eat?", "Banana"
Ask "Have the best computer magazine?", "PC Advisor"

MsgBox "You scored " & RightAnswers & " out of " & Title

Sub Ask(question, answer)
    Dim msg
    msg = InputBox(question, "Question")
    If StrComp(msg, answer, vbTextCompare) = 0 Then
        MsgBox "Right!", vbInformation, Title
        RightAnswers = RightAnswers + 1
    Else
        MsgBox "Wrong!", vbExclamation, Title
    End If
End Sub
    
```

1 Using Notepad or any plain text editor (we've used VBS Editor which is available on this month's cover disc), type in the code shown right. To avoid duplicating code we've written a sub-routine, Ask, which asks the question and compares the response with the correct answer. It displays a message to say whether the answer was right or wrong and keeps a running total of the number of right answers

2 Save the code you entered as quiz.vbs. To run the script you simply double-click the VBS file. If you made a mistake, you'll get an error message showing the line and character position where the error was found. Correct the mistake and try again. If all is well, the program will ask a series of questions, as shown right



64KB of RAM installed) programmers identified years using only their last two digits. But some of these programs had a much longer life than originally intended and, as the 1990s drew to a close, fears rose about the consequences of the year changing from 99 to 00. Yes, we're talking about the Millennium bug.

Development of algorithmic language, otherwise known as Algol, begun in 1958. Though not used today, Algol pioneered concepts such as block structure and the use of formal grammar to define the syntax of the language: principles that are used in current programming languages.

The language Pascal (named after the mathematician Blaise Pascal) is perhaps

the closest derivative of Algol. It became popular in the 1980s thanks to Borland's small and ultra-fast Turbo Pascal compiler and is the language behind the company's current Delphi product.

The other major derivative of Algol was a language called C, which was developed in 1972. This was much more cryptic than Pascal and therefore difficult for programmers to read. It was also less strict – the compiler undertakes fewer checks, giving more scope for errors.

Nevertheless, C became the most popular programming language because C was used in the Unix operating system and widely taught in universities.

OOPs upside your head

C++, first published in 1983, addressed some of the shortcomings of C. It added support for OOP (object-oriented programming), a technique that allowed complex code to be encapsulated in simple-to-use building blocks called 'objects'. Most of Microsoft Windows – and most of the company's other applications – are written in C++.

Basic, which stands for beginner's all-purpose symbolic instruction code, was developed in 1964 as a language to teach programming. Instead of a compiler, Basic used an interpreter that translated program statements to machine code and executed them a line at a time.

Basic was commonly used in the first home computers that appeared during the 1970s.

Microsoft was formed in 1974 when Bill Gates and his pal Paul Allen bunked off college in order to develop a Basic interpreter for an early home PC, the Altair.

In 1991 the company released Visual Basic, a tool intended to simplify the development of Windows programs with a graphical interface – something that was difficult in C. It has now grown into a powerful programming tool. Other variants of Microsoft Basic include Visual Basic for Applications, which is supported by Office programs, and VBScript, a version that can be embedded within web pages.

Visual Basic programmers start with a blank window or form and create a graphical interface by dropping buttons, edit boxes and other elements on to it.

They then write short blocks of code that are executed whenever an event occurs – for instance, when a menu item is selected. Borland used the same model in its Visual Pascal product, renamed Delphi upon its release in 1994.

Java, another C derivative, was released by Sun Microsystems in 1995. Designed so that it would not be tied to any specific hardware, Java was seen as a tool that would lead to the development of portable applications that would run on any computer. This hasn't happened as Java programs run more slowly and consume more memory than those written for a specific platform.

However, Java is now widely used as a scripting language for web pages that will run in different browsers on a variety of computers. ■