

of course[®]

VOLUME 2, NUMBER 1

WINTER / SPRING 2002

Brand new virtual machine

Parrot Uncaged!



The latest versions
Perl • Java
PHP • MySQL
Tcl/Tk • Expect

New Year Brings Euro Change



Hello, and welcome to Well House Consultant's first "Of Course" Newsletter of 2002.

On the 1st of January, the Euro replaced 12 national currencies in Europe. We were in Austria for the new year and marvelled at the smooth transition. On the right hand side of this page, you'll find an article that lists the alternative ways you can provide dual pricing on your own web site, and if you look to our course schedule on page 15, you'll find that we now quote fixed prices in both Euros and pounds.

At Well House Consultants, we provide training courses in Perl, PHP, Tcl, MySQL and Java, running public and private courses at our own training centre in Wiltshire. We also train on site in your offices if you prefer. All of our courses are presented and written by our own staff, who are professional practitioners of the subjects on which they teach. And having students with such varied backgrounds and experience is the reason why we offer so many courses. Find your course needs on pages 6 and 7.

The long-awaited Perl 6 will incorporate a brand new virtual machine, written with the benefit of experience of the old bytecode machine. This new virtual machine is known as "parrot", and it's designed to suit not only Perl 6, but also Perl 5, Python and Tcl. We discuss this topic beginning on page 8.

We use this newsletter to keep past trainees up-to-date on the latest news on Perl, PHP, Java and other topics on which we train, so you'll find current release information and news of Perl 6. Refer to page 13 for the latest open source releases.

Our tutor or administrator will be more than happy to discuss which courses are relevant to you, prices, availability, and other details. There's no hard sell; the majority of our business is repeated from client companies who have used us previously and like our product.

— Graham J. Ellis

Star performance for Win32

Want a server on which to develop, but don't want the hassle of sourcing and installing all the bits you need?

If you're developing Web applications in Perl, and your desktop machine is running a Windows operating system, you might like to have a look at IndigoPerl.

IndigoPerl is a binary build of Perl 5.6 for Win32 with an integrated Apache web server for testing and developing CGI scripts. Top of its feature list comes "easy installation". You should be able to have a test and development server on desktop in very short order.

IndigoPerl is distributed by IndigoSTAR software, and can be downloaded under their free software license from <http://www.indigostar.com/indigoperl.htm>.



When is a "modem" not a modem?

Most connect a computer to the Internet via an analogue telephone line through a modem.

A modem got its name from what it does as a MODulator / DEModulator, a device that turns a computer's digital pulses into audio frequencies for the telephone line and vice versa. But the term "modem" has evolved into other uses these days.

If you use an ISDN line, which is a digital telephone line, you may still call

connection hardware a "modem", but it will be neither modulating nor demodulating signals. A "cable modem" accessing the Internet through broadband is another digital connection, but really acts as a convertor.

Wireless networking technology such as IEEE 802.11b converts digital signals into radio frequencies, yet wireless network cards and base station are still referred to as modems.

Consider putting all your cookies in one place

The European Parliament has voted for a regulation of cookies as part of their adoption of the Cybercrime treaty. Under the regulation, it would be illegal to "use Cookies, Spyware, web bugs and other hidden identifiers ... to store information or trace the activities of the user".

There is currently some dispute amongst computer professionals as to what this will mean in practice; it seems that under the proposals, undated cookies (those that die when you close your browser, and are not stored to your disk) are still acceptable, but persistent cookies are not. It's not clear whether persistent cookies may legally be sent from a server in a country outside Europe (e.g. the USA) to a browser in Europe, or whether browser authors should supply European

versions that don't allow cookies. It does seem that persistent cookies are allowed on a web site provided that the web site asks the user whether he or she will accept them.

Until these matters are resolved, we suggest that you keep all the code that relates to cookies and hidden fields in one "module" or "class"; you can change sites easily to reflect changing legal needs. Modular design such as this is, in any case, very good practice.

How much risk in fixing the exchange rate?

The Euro-to-Pound rate does fluctuate. Here are the rates quoted as we write this article, compared to what the rates were one, two and three years ago.

Jan2002	£ 0.6169
Jan2001	£ 0.6334
Jan2000	£ 0.6064
Jan1999	£ 0.6983



Euro choice: a few options

With the Euro now in use as the day-to-day currency of most of Europe, you might if you should quote prices and accept payments in Euros, even if you haven't quoted in Pounds, Francs and Lira before.

The exchange rate between the Pound and the Euro fluctuates, and you have a choice to make if you're moving to a dual currency web site. You could:

- Update the exchange rate on your web site daily (you can get the latest figure from <http://www.ecb.int>), and quote Euro prices marked "for guidance only", invoicing customers at the rate that holds on the day they order or have the goods dispatched.
- Update the exchange rate daily, and issue quotes in both currencies. You'll need to keep good records of quotes, as prices will vary on a daily basis.
- Update the exchange rate less frequently. You'll be taking a risk that you don't lose a lot of money on fluctuations, but your customers will appreciate being offered prices that don't change every day, and they will be much happier to place Euro orders knowing that the price won't change while they're actually faxing the order through.
- Set fixed prices for your products in both Euros and Pounds, based on the current exchange rate, and review both sets of prices at the same time on an annual basis. You'll be able to fix individual prices in both currencies so that you have easy-to-follow prices and not nastily rounded ones.

If you accept credit card payments, check with your credit card company. You will usually have to charge UK prices in Sterling. If your client has a Euro card, the company will do the conversion so that payment is in Euros at the current rate.

All public courses are run at our specially fitted training centre in Melksham, Wiltshire.

Each trainee on our public course has exclusive use of a laptop computer for the course duration, including access to our servers and broadband Internet access. Public courses are limited to a maximum of eight trainees, and are presented by our own staff who write the material themselves, and who have practical experience in its application.

Melksham is a town in Wiltshire, about 100 miles west of London. It's easily reached by train (86 minutes from London's Paddington station) or road (just a few miles off the M4 motorway).



Poseidon Temple, Lindos, Rhodes. Cadmus, king of Phoenicia, built this temple to Poseidon and dedicated a bronze cauldron with an inscription in Phoenician script to Athena. This is Mythology's way of explaining the introduction of the Phoenician alphabet.

A story of gods and mortals

Once upon a time, a long long time ago, all was not well with the Greek Gods.

Zeus and Poseidon, Athena and Apollo, Aphrodite and Hermes all found themselves spending too much time managing Olympus and too little time involved in the godly life of ruling the world and, of course, partying.

Why the problem? Staff! Someone had to do the cooking and the washing up. Excellent chefs they employed, but those chefs were mortals forever complaining (are chefs not said to be temperamental to this day?), and for even having to be replaced for one reason or another!

Drastic problems required drastic actions, and Zeus and the other 11 Olympian gods took such an action. They took the best of the best of their mortal staff and struck a deal with them. Henceforth, those mortals were to be immortal and be available at the beck and call of the gods, to provide their service to the gods (be it cooking or ironing, dusting or serving at table) at any time the service was needed, without complaint, and only troubling the gods with any issue if it was a serious problem.

These lower ranking immortals were known as the "daemons".

Unix / Linux: A daemon is a background process, running whenever the computer hosting it is running, and providing services to users who call for that service. Examples of daemons include web server programs, print spool managers, and database engines, with names like *httpd*, *lpd* and *mysqld*.

You can build the MySQL database engine into your application as an alternative to running it as a daemon. Embedded MySQL is ideal for use in Internet applications, turnkey solutions, and more.

We provide training on Tcl, Tk and Expect because they have stood the test of time. Cameron Laird, who writes frequently on computing technologies and standards, illustrates this in his recent post to the comp.lang.tcl newsgroup:

"Tcl is the single best language SysAds can learn. Scotty is the best open-source SNMP package; it's simply indispensable (even though largely unmaintained for a couple of years). Expect is an even greater necessity for SysAds ... no other language offers anything nearly as

Extracting text from a Word document to put into Perl

Perl embraces new technologies and standards by providing a mechanism through which they can be supported. They are either built in to the Perl language itself, or provided through modules.

For example: What if you want to take the text from some Word documents and have that text available in a different format on your web server? The natural choice is Perl and the hardest part of such a task is usually finding out how to do it, which modules to load, and from where, and how to call them.

Here's a piece of code for extracting text from a Word document:

```
use Win32::OLE;
use Win32::OLE::Enum;

$document = Win32::OLE -> GetObject($ARGV[1]);
open (FH,">$ARGV[0]");

$paragraphs = $document->Paragraphs();
$enumerate = new Win32::OLE::Enum($paragraphs);
while(defined($paragraph = $enumerate->Next()))
{
    $style = $paragraph->{Style}->{NameLocal};
    print FH "+$style\n";
    $text = $paragraph->{Range}->{Text};
    $text =~ s/[\n\r]//g;
    $text =~ s/\x0b/\n/g;
    print FH "$text\n";
}
```

This example uses the Win32 modules to access the Word document (named as the second parameter of the command line) and saves each paragraph style name and contents into a plain text file (named as the first parameter on the command line).

The Win32 modules are available on the CPAN, and are also a standard part of the ActiveState distribution. They make calls to Word itself, so this Perl application must be run on a Windows machine, and then the extracted data transferred.

Keeping to the basics because they work

convenient. Finally, Tcl/Tk gives the easiest, most natural way to build the tiny little control panels Leam was describing, and that can revolutionise daily life for SysAds. In principle, Perl/Tk and Tkinter (and comparable for Ruby, Scheme, ...) have all the same functionality, and even more ... but they just aren't as easy to install and deploy. TclKit gives all the advantages of single-file deployment. In everything else – database access, 'net protocols, maintainability – reasonable people can argue about who's really on top. Expect, Tcl/Tk, TclKit, and Scotty make

Tcl an easy choice, though. Really. It's just that almost no one knows that, even after the better part of a decade of Expect and Tk proselytization."

And in another posting about Expect: "It has a simply phenomenal record in several regards ... no matter what question people ask Don [Libes], his answer always is (I exaggerate only slightly), "It's in the book." He's always right; somewhere in the book, PUBLISHED OVER SEVEN YEARS AGO, are all the answers to questions Expect programmers have. There are rocks less solid than that.

Open source tools triumph in market-leading database application

by Phil Buckley, MPC Data Limited



Our client needed a forklift truck management information and access control system with sophisticated features such as smart card programming for access control, flexible report generation, automated scheduling of common tasks, remote troubleshooting, foreign language support and free warranty. The product was required to run under all flavours of Windows and be sufficiently scalable to suit single site operators with ten or so forklifts right up to international logistics companies operating over hundreds of sites with tens of thousands of trucks – all this inside three months.

MPC Data are used to working with tight deadlines. Our flexible and co-operative working practices enable us to quickly assemble a team of dedicated professionals to meet our clients' needs. We have strong quality procedures and operate regular, open reporting structures to gather changing requirements and share status with clients. MPC Data has built its own intranet using MySQL and PHP4 to coordinate both systems electronically.

Easy scalability was a critical requirement for any end product that was going to be successful in this marketplace. Our solution had to fit within the budget constraints

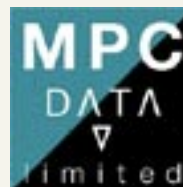
of smaller organizations whilst also delivering the high-end capability needed to meet the requirements of the larger fleet operators. Clearly the larger systems pointed towards the use of a relational database but the use of commercially available tools would place the end product beyond the means of many organizations.

Solution Based on Open Source Tools

Having settled on Crystal Reports to tackle the requirement for flexible reporting, we needed a high-end database server with an SQL interface at a low-end cost. MySQL seemed like the obvious choice. It could be bundled with the product for less than 10% of the license fee of a commercially available relational database server. With the addition of the MyODBC interface the application code could make use of standardized SQL – a key advantage in allowing larger operators to switch to pre-established servers such as Oracle or SQL Server without source code modifications. By enabling InnoDB table support the database engine can confidently handle complex, transaction-safe multiple table updates from concurrent users for very little coding effort.

The project team depended on the CVS version control system to

track changes during evolving design and implementation phases as the customer introduced fresh ideas. The user interface was developed using Microsoft® Visual C++ and MFC to provide a familiar, feature rich application framework easily picked-up by users with limited computer skills. Database classes can be derived from MFC record-set classes and quickly implemented visually with the class wizard. The wizard creates much of the boilerplate class code by querying the table structure through the MyODBC interface.



Creative software solutions for product development and technical business problems

MPC Data is a MySQL Solution Provider with a 17-year track record in software development.

We also provide training and consultancy for CVS. A detailed case study of this project and many others are available on request.

Contact the author

pbuckley@mpc-data.co.uk
or visit www.mpc-data.co.uk

More dynamic than an array?

Most traditional programming languages include support for arrays, a whole series of pieces of information of the same type. The elements of an array are usually held in successive memory locations, and each element takes the same amount of storage.

With languages such as C, the number of elements in an array (its dimension) was defined at compile time, and was then fixed in the executable program file. This was a problem if the programmer didn't know the maximum size, and mechanisms such as "malloc" were added to allow an array to be defined while the program ran.

In contrast to C, Java has a dynamic memory allocation model, so arrays can be dimensioned at run time. You can write a program that works out how big an array needs to be and then dimension it to allocate the memory. Alas, once an array is dimensioned, you can't come back later and simply extend it.

Java supports a number of collection types (via standard classes) that let you define structures to hold a whole series of objects, even if you don't know how many objects you'll want to hold. Vectors (known as ArrayLists in Java 2) can hold as many objects as you like, and you don't have to tell Java how many. The information isn't stored sequentially in memory, so they aren't as efficient as an array. They're not a part of the base Java language so you have to access them through an object syntax, but they do solve the problems of changing an array size even after it has some data stored in it.

Array elements are traditionally numbered (indexed) from 0 or 1 up. Java's standard HashTable class (also a HashMap from Java 2) lets you give each element a unique name, though it brings in a whole series of new considerations if you need to be able to iterate through all the elements in such a collection.

Perl's lists are the equivalent of a Vector in Java, and "hashes" are the equivalent of a HashTable. Perl doesn't actually use arrays even though the word encompasses lists and hashes.

In Tcl, an array is the equivalent of a HashTable in Java. As all data in Tcl is held in strings, there is no equivalent to an array or a Vector in Java; elements can't be counted like that.

PHP arrays can be treated as either HashTables or Vectors, and can be dynamically extended. You can reference them by number (position) or by text key as you wish.



Melksham spa bottles, thought to be one-of-a-kind. (Above) a close view of the older bottle's "label"

In September 1988, Isabel Ide wrote a booklet, "The Celebrated Melksham Spa". It is a fascinating account shedding much light onto the history of these curious homes, as in this excerpt:

"There appears to have been a belief that the nastier the waters tasted the more efficacious they were; the springs at Melksham seem to have been nasty enough to satisfy this requirement. Dr Gibbes of Bath recommended the Chalybeate Well for both bilious and scorbutic complaints. The saline waters were considered to be a natural purgative, which was necessary in those days of large meals and heavy drinking."

From a watery beginning...

Our Melksham training centre was built in Georgian times (1814), one of a group of six lodging houses for visitors to the town. An unknown source in the 1840s wrote:

"Two mineral springs, one Saline and the other Chalybeate, were discovered in the vicinity some years ago; and in 1816 a new saline spa was obtained by drilling to a depth of 351 feet. It was anticipated that Melksham would advance into note, in consequence of the magical properties these waters possessed, and every accommodation for visitors was provided, including a pump room, hot and cold basins, a handsome crescent, an agreeable promenade.

But with all these inducements, success was not commanded."

The waters were bottled and sold until late Victorian times, however.

The bottles are very rare – we've

been able to find out about just two, and last Autumn their owner (a bottle collector who lives in Bristol) brought them over to show us.

The older stone bottle was made by Price in Bristol, and is hard to date as Price's factory and all their records were destroyed by fire. We're told that it's probably quite early, being a single stone colour rather than having a darker brown glaze as you would see in later stone bottles. The other bottle is a much later glass bottle, probably from later Victorian times. It probably lay in someone's shed for years as the label (which is what identifies it to Melksham) has been chewed by mice.

It would have been marvellous to have found a bottle, or even pieces, in our garden, but the rubbish buried there was much more modern; certainly the old car wheel and tyre could be dated to the late 20th century.

PHP security improvement in latest release (v 4.1.0)

PHP has come from nothing in the last two years. Why? Because it's superbly suited for server side executable use.

It's one of the few languages designed for that specific use, and that shows in the superb range of built-in functions, and the ease with which server side code can be written. No longer does the programmer need to learn a language and an interfacing scheme (such as CGI); it's all there in PHP.

PHP is also easy to code. Some would argue that it's too easy. Create a form, have a user complete that form on his browser, and everything that he's entered magically appears as variables in your PHP.

Fabulous, but there's a potential security risk there. From our very first PHP course, we warned authors that they should initialise all variables that don't come through as form (or cookie, or session) inputs. Why? Simply because an unscrupulous user with appropriate knowledge could maliciously initialise an unset PHP variable by modifying a form.

As of the latest release of PHP (4.1.0, in December 2001), there's a brand new series of arrays (called `$_GET`, `$_POST`, `$_COOKIE` etc.) from which you're encouraged to read your form, cookie, or environment inputs.

The potential confusion between your own variables and form variables

is cleared.

Of course, the vast majority of PHP applications out there use the old mechanism which remains available, although deprecated. For new installations with new code, you're encouraged to turn off `register_globals` in the configuration file `php.ini`, and in time the older mechanism will fade away.

Stop the press!

PHP 4.1.1 was released on 26th December 2001. It contains a minor bug fix release.

Other changes released in PHP 4.1.0 include:

- Highly improved performance, especially under Windows
- Support for "versioning" of extensions
- Output compression support
- Lots of new functions



PHP provides the power behind more and more web sites, just as these magnificent "Shay" steam engines provide the power behind trains on the Cass Scenic Railroad in West Virginia".

TIME LINE – Well House Consultants Ltd.

We specialise in open source language training. did you realise that we've been teaching Perl for five years now, and have a range of 12 different public courses?

1995 Well House Consultants formed to provide Unix (Solaris) training for other training suppliers, and to provide internet and web services for those suppliers.

1996 We wrote and presented our first Perl course, selling it through the catalogue of one of our customers, and running it at their training centre.

1997 Java courses were introduced, and our Perl range was extended. We were now teaching very few Unix courses as all our time was spent

specialising in Perl and Java, and working on web sites.

1998 We took over the formatting and printing of our own training notes, introduced our "quick reference cards" and provided online access to course examples for the first time.

1999 Our office was getting cramped! We bought a Georgian lodging house on four floors in Melksham – "In need of some renovation" as estate agent's details might say – and started work on it!

2000 In January, our office opened in Melksham, and our public courses also transferred there in the Autumn. Since we'd had the opportunity to furnish from scratch, our training

centre had infrastructure needed for the type of courses we run, and was equipped appropriately. 2000 also saw the introduction of our Tcl courses.

2001 The first edition of our "Of Course" newsletter was published at the start of the year, and we continued to improve our courses and infrastructure, with facilities such as broadband access to all trainee desktops, wireless networking, and



the doubling in size of our technical library. PHP and MySQL courses were introduced in the summer, and we got planning

consent to use more of the building for the business than we had envisioned at first.

2002 We'll be looking to improve our service still further. We now accept payment in Euros as well as Sterling and USD, and we're now testing our secure server to start accepting your bookings on line. We've already updated two of our courses in the first few days of the year, and more updates will follow as we continue to track technology.

And we've held our prices. A five-day public course still costs only £1,090 – that's actually less than it was two years ago (£1,215) when we had an extra level of reseller in the loop.

Decisions Decisions



Question: “Why do you have more than one course on subjects like Perl and Tcl?”

Answer: Because we specialise in the subjects we teach; we know that trainees will come to us with different backgrounds, and will have different objectives from the course. We want you to choose a course or series of courses to suit your needs.

This article takes you through each of the subjects on which we run regular public courses and helps you select which course(s) will be most relevant to your requirements.

If you have a group of trainees all requiring the same course at the same time, please let us know; we can run a special course for you at your office (we provide all the equipment) or at our training centre, and that course can be tailored to suit your needs.

Perl

Starter courses

If you've never programmed before, or you're very rusty, you'll want to start on our 5 day [Learning to Program in Perl](#) course. On the other hand, if you're already an experienced programmer in another language, you should start with the four-day [Perl Programming](#) course.

More advanced courses

For users who'll be using Perl for straight-forward data manipulation and reporting tasks, our starter courses are sufficient.

If you'll be using Perl in a Web and network environment, then you'll want to follow your starter course with [Using Perl on the Web](#), which covers subjects such as server side programming, Perl's interfaces to Telnet and FTP, etc. Using Perl on the Web is a two-day course.

If you're going to be writing and maintaining larger applications in Perl, you'll probably want to follow your starter course with a three-day [Perl for Larger Projects](#) course, which covers subjects such as handling huge amounts of data, database interfacing, writing objects in Perl, and more.

There's very little overlap between Perl on the Web and Perl for Larger Projects, so if you're going to be writing a major web-based application in Perl, you should book onto both courses. We always schedule these two courses in the same week (two followed by three days), so that you can stay on.

If you'll be using Perl to access MySQL relational databases, but you don't know about MySQL yet, you may wish to attend our MySQL course.

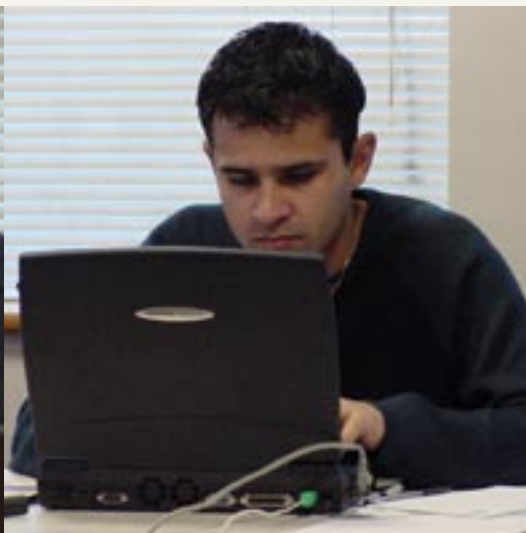
What if you want something more specialised? For example you want to learn about the SNMP interface from Perl, or about Perl/Tk. We have other, more specialised, training material available that we use as appropriate on private courses.

If you've attended one of our regular public courses, but need to learn more on a specialist topic, or you want to troubleshoot a specific project you have, we'll run a special one-on-one day for you at public course rates to cover the extra material. We call this the [Perl Extra](#).

Java

Starter courses

Java has an very wide range of uses – from programming telephones through to enterprise applications. All Java programmers will require a core knowledge of the language, as covered on our [Java Programming for the Web](#) course, which



run for five days.

The Java Programming for the Web course assumes some prior programming experience. We find that the majority of trainees are looking for a conversion to Java rather than to learn to program from scratch in the language. But, we can also help you if you have no prior experience. Please ask, and we might be able to provide a custom extra day for you prior to the main course. This introduction can be counted as a Java Extra day.

More advanced courses

Certain more advanced topics such as servlets, applets and JSPs are introduced in our Java Programming for the Web course, so it might well be that the one course provides all the Java training you need to get you going.

We have a wide range of further advanced training material available, which we normally use for on-site or private courses. If you've attended our regular public Java course and want to cover some of these extra topics, we'll run a special one-on-one day for you at public course rates to cover the extra material on [Java Extra](#).

If you'll be using Java to access MySQL relational databases, but you don't know about MySQL yet, you may wish to attend our MySQL course..

Tcl, Tcl/Tk and Expect

Tcl is a scripting language that can be embedded within an application. We can provide training on the Tcl language itself, and also on the Tk extension which provides a programmable Graphic User Interface (GUI), and on Expect, which lets you automate processes designed for use through an interactive (i.e. keyboard) interface.

Our three-day [Tcl Basics](#) teaches you the Tcl language, and also covers the extras that are provided in Expect; prior programming knowledge is useful, but Tcl differs greatly from other programming languages and sometimes prior experience of another language can actually be a disadvantage as it leads you to false assumptions.

If you're going to be using the Graphic User Interface provided by Tk, stay on for our two-day [Tcl/Tk](#) course.

PHP

Starter courses

PHP is a server side, HTML embedded scripting language.

If you're not familiar with server side programming, the structure of the web, or HTML, then you should attend our one-day [Technology for PHP](#) course to introduce you to these background topics.

More advanced courses

Our three-day [PHP Programming](#) course follows on from Technology for PHP and covers the PHP language, the wide range of built-in functions it includes, and how to use PHP to write complete server side applications.

If you'll be using PHP to access MySQL relational databases, but you don't

Pricing

We keep our pricing structure simple. No matter which courses you choose, the price is £290 (≈470) for the first day and £200 (≈330) for subsequent days of each course.

Days	Pounds	Euros
1	290.00	470.00
2	490.00	800.00
3	690.00	1,130.00
4	890.00	1,460.00
5	1,090.00	1,790.00

If you book several trainees onto the same course at the same time, you will save £90 (≈140) on the second and subsequent trainee, but beyond that we don't offer discounts – our list prices are already competitive.

Prices quoted exclude VAT.

If you have a group of four or more trainees requiring the same training at the same time, please enquire about custom courses which can be run at our training centre or at your offices. Such courses are tailored to meet your exact needs, and will be more cost-effective than taking up many places on a public course. We have an easy-to use online form that will give you a no-obligation quote. You simply type in the post code of the course location. form is one click off our home page at www.wellho.net, or type in: <http://www.wellho.net/course/otc.html>

know about MySQL yet, you may wish to attend our MySQL course.

MySQL

MySQL is described as "the World's most popular open source database". Our two-day [MySQL](#) course covers relational database design, Structured Query Language (SQL) as used in MySQL, and MySQL installation and administration.

If you're going to be using MySQL as your database in a PHP, Java or Perl programming environment, you should learn that programming language before you attend the MySQL course. Appropriate courses are described in other sections of this article.

We recently provided a customised Java course for Risk Decisions, whose "Predict!" risk management software is used by a wide range of Blue Chip companies.

Pictured on these two pages, anticlockwise from the left are:

Sasha Souchtchenko

Nigel Vereker

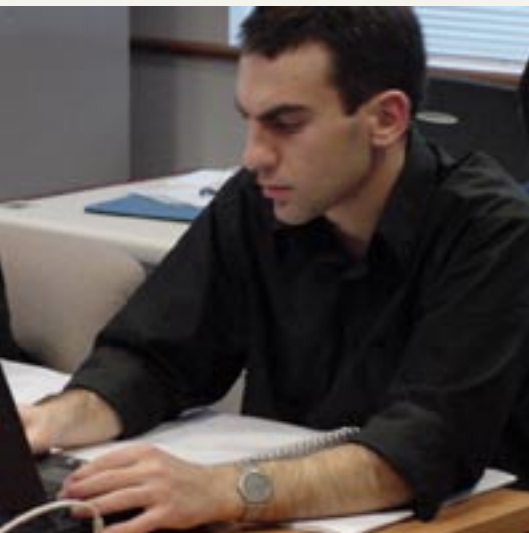
Susheel Chumber

Michael Morris with John Burgess behind and to the right

Yannis Fitsakis

Vincci Lau

Aleks Sienkiewicz



Getting your applications to work

Compiled
Interpretative
Virtual Machine

Whatever programming language they use, you want your programming staff to be able to develop and maintain applications quickly and efficiently, and you want the applications to run fast. With traditional languages, however, there had to be a trade-off between these two goals.

Compiled languages

Computer languages such as C and C++ (and Cobol and Fortran and many many more) are what we call "compiled" languages. The programmer edits or enters his program into a series of source files, then runs it through a compiler that produces "object files", snippets of executable program suitable for use on a specific type of processor and operating system. A further stage, the loader (or linker or taskbuilder) connects all the object files necessary to make a complete application, together with standard libraries, into an executable program file.

Compiled languages can result in programs that run very fast indeed on their target computer, but the need to re-compile and re-load, and have different program variants depending on operating system and processor, means that the development process can be long-winded. Tools such as "make" help, but a traditional compiled language will never be as easy as "change and run".

Interpretive languages

For applications which are to be written rapidly and don't need to run fast, you would traditionally use an interpretive language. Many versions of Basic are interpretive, as are Shell programming languages, batch files, and so on. With an interpretive language, a program called the interpreter runs every time that the application is to run, and it interprets and then acts on the application line-by-line.

Developing a program in an interpretive language takes away all the hassle of compilers and loaders; just point the interpreter at the text file containing the program source and it will run. But it may run slowly. Each line has to be interpreted each time it is to be performed, and so you have extra run time overheads that make interpretive language code execution an order of magnitude slower than compiled code execution.

Virtual machine-based languages

Compiled languages are fast to run, and interpretive languages are simpler to develop. But compiled languages are more complex to develop, and interpretive languages run much slower. Is there another scheme that gives us the best of both worlds? Not quite, but we can achieve something that runs much faster than a pure interpretive system, and is easier to develop than a pure compiled system, using the concept of a virtual machine.

A virtual machine-based language has both a compiler and an interpreter. The programmer enters his code into a plain text file (exactly as he would for a compiled or an interpretive system), then presents it to the compiler. Rather than this compiler producing machine code for a particular operating system and processor, it produces machine code for a "virtual machine".

The virtual machine is run after the compiler; it takes the machine code output from the compiler and interprets and runs that machine code. The virtual machine doesn't have to do all the program analysis of a regular interpreter (that's already been done by the compiler), so it can run very much faster than such an interpreter.

One of the earliest virtual machine-based systems was the P-System of UCSD (University of California at San Diego) and their Pascal compiler. The P-codes (as the intermediate compiler output was known) could be run through virtual machines on a huge range of computers with DEC, Zilog, Motorola, Intel and many other processor types. A similar concept was also adopted by Larry Wall when he created the Perl language, and more recently it has been adopted in Java and in Microsoft's .NET environment. PHP also uses an embedded "compile then execute" scheme from release 4 on, through the Zend engine.

Perl's Virtual Machine

From its early days, Perl used a virtual machine. In the Perl model (up to and including release 5 of Perl), the compiler and interpreter are built into the same

continued on page 10



Parrot

Perl 6's virtual machine

The forthcoming Perl 6 will incorporate a brand new virtual machine, written with the benefit of experience of the old bytecode machine as well as other virtual machines, such as the JVM. This new virtual machine is known as "parrot", and it's designed to suit not only Perl 6, but also Perl 5, and probably other bytecode compiled languages such as Python and Tcl.

For the more technical reader, Parrot is a register-based virtual machine that supports dynamic data typing by handling Perl's ability to switch variables from strings to numbers to references rather than having to emulate such changes more slowly within software. It's planned that the Perl 6 compiler will be able to produce output for a JVM and for .NET, but the code will run more slowly than it will on Parrot.

Parrot's input is a compact binary byte format, "parrot byte code". Unlike Perl 5, where the byte code was purely internal to Perl until B::Bytecode came along, this new byte code is designed for saving to file and distribution via a network, etc., and the file extension you'll most commonly see will be .pbc.

Usually, you write your source code in a language such as Java or Perl, and have the compiler for that language produce the appropriate byte code. Although byte code formats are published, the application developer and maintainer will rarely access them directly.

It is instructional, though, for more experienced programmers to have an understanding of byte codes and how they work. The early test releases of Parrot, which you can download from the CPAN, come with an assembler program that lets you code Parrot virtual machine instructions into a text file, and then convert them to a .pbc (Perl Byte Code) file.

Here's a sample program written in Parrot assembler:

```
# This is the first test of Parrot
print "Hello birdie World - triangle numbers\n"
print "This is a test of Parrot\n"
set I1,0
set I2,0
REDO: inc I2
add I1, I2, I1
print I2
print " gives "
print I1
print "\n"
lt I2, 10, REDO
end
```

We convert that into a Parrot byte code file using the assembler program (itself written in Perl):

```
$ perl assemble.pl hello.par > hello.pbc
$
```


There's nothing to stop you looking inside the .pbc file provided you use an appropriate tool; this example looks at the file as text characters, and also as four byte integers

```
$ od -c hello.pbc
00000000  i  U  1  001  \0  \0  \0  \0  '  \0  \0  \0  \0  004  \0  \0  \0
00000020  s  \0  \0  \0  \0  8  \0  \0  \0  \0  \0  \0  \0  \0  \0  \0  \0
00000040  \0  \0  \0  \0  \0  &  \0  \0  \0  H  e  l  l  o  b  i
00000060  r  d  i  e  W  o  r  l  d  -  t  r  i
0000100  a  n  g  l  e  n  u  m  b  e  r  s  \n  \0  \0
0000120  s  \0  \0  \0  \0  ,  \0  \0  \0  \0  \0  \0  \0  \0  \0
0000140  \0  \0  \0  \0  031  \0  \0  \0  T  h  i  s  i  s
0000160  a  t  e  s  t  o  f  P  a  r  r  o  t
0000200  \n  \0  \0  \0  \0  s  \0  \0  \0  030  \0  \0  \0  \0  \0  \0
0000220  \0  \0  \0  \0  \0  \0  \0  \0  \0  \a  \0  \0  \0  \0  \0  \0
0000240  e  s  \0  s  \0  \0  \0  024  \0  \0  \0  \0  \0  \0  \0
0000260  \0  \0  \0  \0  \0  \0  \0  \0  \0  \0  \0  \0  \n  \0  \0  \0
0000300  t  \0  \0  \0  \0  037  \0  \0  \0  \0  \0  \0  \0  037  \0  \0  \0
0000320  001  \0  \0  \0  \0  8  \0  \0  \0  \0  \0  \0  \0  \0  \0  \0  \0
0000340  8  \0  \0  \0  \0  002  \0  \0  \0  \0  \0  \0  \0  "  \0  \0  \0
0000360  002  \0  \0  \0  \0  232  \0  \0  \0  \0  \0  \0  \0  002  \0  \0  \0
0000400  001  \0  \0  \0  \0  032  \0  \0  \0  \0  \0  \0  \0  037  \0  \0  \0
0000420  002  \0  \0  \0  \0  032  \0  \0  \0  \0  \0  \0  \0  037  \0  \0  \0
0000440  003  \0  \0  \0  \0  x  \0  \0  \0  \0  \0  \0  \0  \n  \0  \0  \0
0000460  ò  ŷ  ŷ  ŷ  \0  \0  \0  \0
0000470

$ od -l hello.pbc
00000000  20010401  0  180  4
00000020  115  56  0  0
00000040  0  38  1819043144  1768038511
00000060  1701405810  1919899424  757097580  1769108512
00000100  1818717793  1970151525  1919246957  2675
00000120  115  44  0  0
00000140  0  25  1936287828  544434464
00000160  1702109281  1864397939  1632641126  1953460850
00000200  10  115  24  0
00000220  0  0  7  1986619168
00000240  2126693  115  20  0
00000260  0  0  1  10
00000300  116  31  0  31
00000320  1  56  1  0
00000340  56  2  0  178
00000360  2  154  1  2
00000400  1  26  2  31
00000420  2  26  1  31
00000440  3  120  2  10
00000460  -14  0
0000470
```

There's also a disassembler supplied with Parrot. Not only does this help you understand what's what in the byte code, but it also helps compiler writers and the more advanced developers see the code that's actually generated by their applications. Here's our program disassembled; if you study this in association with the output dumps and original source code, you'll start to get a flavour of how Parrot works.

And we can then run the program through the parrot virtual machine:

```
$ parrot hello.pbc
Hello birdie World -
triangle numbers
This is a test of Parrot
1 gives 1
2 gives 3
3 gives 6
4 gives 10
5 gives 15
6 gives 21
7 gives 28
8 gives 36
9 gives 45
10 gives 55
$
```

```
#
# Disassembly of Parrot Byte Code from 'hello.pbc'
#
# Segments:
#
# * Magic Number:      4 bytes
# * Fixup Table:       0 bytes
# * Const Table:      180 bytes
# * Byte Code:        116 bytes (29 opcode_ts)
#
# Constant  Type      Data
# -----
# 0  PFC_STRING  "Hello birdie World - triangle numbers\n"
# 1  PFC_STRING  "This is a test of Parrot\n"
# 2  PFC_STRING  " gives "
# 3  PFC_STRING  "\n"
#
# WORD  BYTE  BYTE CODE  LABEL  OPERATION  ARGUMENTS
# -----
00000000 [00000000]: 00000031 00000000  print  [sc:0]
00000002 [00000008]: 00000031 00000001  print  [sc:1]
00000004 [00000016]: 00000056 00000001 00000000  set  I1, 0
00000007 [00000028]: 00000056 00000002 00000000  set  I2, 0
00000010 [00000040]: 00000178 00000002  inc  I2
00000012 [00000048]: 00000154 00000001 00000002 00000001  add  I1, I2, I1
00000016 [00000064]: 00000026 00000002  print  I2
00000018 [00000072]: 00000031 00000002  print  [sc:2]
00000020 [00000080]: 00000026 00000001  print  I1
00000022 [00000088]: 00000031 00000003  print  [sc:3]
00000024 [00000096]: 00000120 00000002 00000010 -0000014  lt  I2, 10, -14
00000028 [00000112]: 00000000  end
```



BEECHFIELD HOUSE

THE SHAW COUNTRY HOTEL

THE SPA B&B

Local accommodation around Melksham

People attend courses at our Melksham training centre from far and wide, and there's a good range of well priced accommodation available nearby. Whilst we try to ensure the accuracy of these details, we might

not be informed of any changes, so please verify this information with the establishment when you book your own accommodation. Pricing shown is for single occupancy; many offer family and double occupancy facilities.

There are also self catering, caravan and camping facilities nearby, as well as other hotels, guest houses and B&Bs. Melksham Tourist Information Centre (01225 707 424) can provide you with up-to-date details.

1 The Spa B&B 402, The Spa, Melksham SN12 6QL H. Pavey	P: 01225 707 984	distance from centres (mi)		phone in room	restaurant on premises	takes credit cards
		training	town			
		0.1	0.7			
£20.00-25.00 single, one room en-suite, includes breakfast		Large comfortable listed Georgian home. Quiet position with open views. Ample parking.				
4 The Old Manor 48 Spa Road, Melksham SN12 7NY L. Bailey	P&F: 01225 793 803	distance from centres (mi)		phone in room	restaurant on premises	takes credit cards
		training	town			
		0.5	0.3			
£25.00-35.00 single, one room en-suite, includes breakfast		Set in 3/4 acre, delightful gardens; large English breakfasts a speciality; lovely decor.				
7 The Kings Arms Hotel 20 Market Place, Melksham SN12 6EX	P: 01225 707 272	distance from centres (mi)		phone in room	restaurant on premises	takes credit cards
		training	town			
		0.8	0.0			
£35.00 single, all rooms en-suite + £6.95 full/£4.95 continental bkfst		Recently completely renovated; located in the heart of town.				
15 Conigre Farm Semington Road, Melksham SN12 6BZ D. Edwards	P: 01225 702 229 F: 01225 707 392 W: www.cfhotel.co.uk	distance from centres (mi)		phone in room	restaurant on premises	takes credit cards
		training	town			
		0.9	0.1			
£43.00-52.00 single, all rooms en-suite, SkyTV, includes breakfast		Family-run hotel and restaurant offers finest foods, quality accommodation and friendly environment.				
38 Longhope Guest House 9 Beanacre Road, Melksham SN12 8AG D. Hyatt	P&F: 01225 706 737 E: longhope@aol.com	distance from centres (mi)		phone in room	restaurant on premises	takes credit cards
		training	town			
		2.0	0.9			
£28.00 single, all rooms en-suite, includes breakfast		Large Victorian house set in its own grounds; warm and friendly, well decorated.				
47 Beechfield House Beanacre, Melksham SN12 7PU E: csm@beechfieldhouse.co.uk	P: 01225 703 700 F: 01225 790 118	distance from centres (mi)		phone in room	restaurant on premises	takes credit cards
		training	town			
		2.7	1.6			
£80.00-90.00 single, all rooms en-suite, includes breakfast		Comfortable country house, 21 en-suite bedrooms, four-poster beds available, heated outdoor swimming pool.				
45 The Shaw Country Hotel Bath Road, Shaw, Melksham SN12 8EF J. Lewis	P: 01225 702 836 F: 01225 790 275	distance from centres (mi)		phone in room	restaurant on premises	takes credit cards
		training	town			
		2.8	1.7			
£46.00 single, all rooms en-suite, includes breakfast		A 400-year-old farm house, supremely restored. Mulberry Restaurant has a growing reputation for good food.				
51 Woolley Grange Hotel Woolley Green, Bradford-on-Avon N. Chapman	P: 01225 864 705 F: 01225 864 059 E: info@woolleygrange.com	distance from centres (mi)		phone in room	restaurant on premises	takes credit cards
		training	town			
		6.1	Bradford-on-A approx. 1 mile			
from £90.00 single, all rooms en-suite, includes breakfast		Jacobean manor house in open countryside with outdoor heated pool, croquet, tennis, badminton.				
48 Travel Inn, Cepen Park (#267) W. Chippenham Way (M4, J17) Chippenham SN14 6UZ	P: 01249 462 096 F: 01249 461 387 W: www.travelinn.co.uk	distance from centres (mi)		phone in room	restaurant on premises	takes credit cards
		training	town			
		8.6	Chippenham approx. 1 mile			
£41.95 single, all rooms en-suite + £6.00 full/£4.00 continental bkfst		79 rooms with Brewsters and petrol station nearby.				
52 Travelodge (#813) Moto Services, A36 / A350 Bypass Warminster BA12 7RU	P: 01985 219 539 F: 01985 214 380 W: www.travelodge.co.uk	distance from centres (mi)		phone in room	restaurant on premises	takes credit cards
		training	town			
		12.1	Warminster approx. 1 mile			
£49.95 single, all rooms en-suite + breakfast from £4.45		31 rooms with Burger King, Little Chef and petrol station nearby.				

Parrot

continued from previous page

program ("perl"), and when you run perl on a text file the file is interpreted into "byte code" (Larry's name for the intermediate compiler output) which is the passed straight on into the interpreter.

Thus, to run a Perl program all you do is type "perl programname" and the text in the file "programname" is compiled to byte code which is then interpreted through the Perl virtual machine. It's a single step as far as the user is concerned; after all, Perl was designed to be easy to write and easy to run.

In the more recent versions of Perl (it's now up to release 5.6.1), the "B::Bytecode" module allows users to run a compile to produce a file which can be used to feed Perl's virtual machine at a later time, and perhaps on a different computer.

The B::Bytecode module is officially described as experimental. Perl has been around now for over 13 years, and it's not easy to add a facility such as this to such a mature language. When we come on to Perl 6, now under development, the model changes and will be known as "Parrot".

Java Virtual Machines

The whole concept of Java's portability is based on its virtual machine. Source code is entered into .java files, and then compiled into .class files, suitable for use in a Java Virtual Machine, or JVM.

All Java applications make extensive use of standard classes provided by the authors of Java, and very often they'll use classes written by other suppliers too. These classes, in addition to the one written specifically for your application(s), will usually be available resident on the same system on which the JVM is to be run. The JVM, plus these classes, is known as the Java Runtime Environment, or JRE.

Although Java in the form of .java and .class files is a portable language, the JRE (and in particular the JVM part of it) is specific to the processor and operating system that the files are running on, and also to the environment in which they're used. Commonly used JVMs include ones to run applets (this JVM is built into the browser, or added as a plugin), and Servlets and JSPs (this being built into Tomcat, WebSphere or JRun), as well as the JVM supplied with Java itself, and JVMs that are used on enterprise servers. @

SERVICES

Banks

- 35 Barclays
- 10 HSBC
- 21 Lloyds TSB
- 11 NatWest

Emergency

- 2 Hospital
- 17 Police Station

General

- 19 Post Office
- 32 Public Library
- 22 Tourist information

LEISURE

- 18 The Blue Pool
- 3 Christie Miller Sports Centre

FOOD & DRINK

Groceries

- 40 Aldi
- 25 Iceland
- 54 Made in America
- 31 Sainsbury
- 36 Somerfield

Takeaways

- 20 Acropolis
- 14 The Chippy
- 30 Chop Suey House
- 53 Cod Joe's Caribbean Takeaway
- 5 Cornerstone Coffee Shop
- 26 Country Cottage Pasties
- 37 The Fisheries
- 13 Golden Falls Chinese
- 34 Kebab House
- 39 Lee's Fish and Chips
- 27 Melksham Chinese
- 24 Mount Stevens

Restaurants (most do takeaways)

- 8 Big J's Farmer's Cafe
- 49 Brewsters
- 42 cafe inside Leeke's
- 43 Little Chef
- 41 McDonald's
- 23 Melksham Tandoori
- 46 Mulberry Restaurant
- 12 Refa Balti House
- 40 West End Inn

Sampling of Pubs

- 29 The Bear
- 33 Bud's
- 50 Foresters Arms
- 28 The Grapes
- 44 The New Inn
- 9 The Parson's Nose
- 6 The Tavern

HOTELS / B&Bs

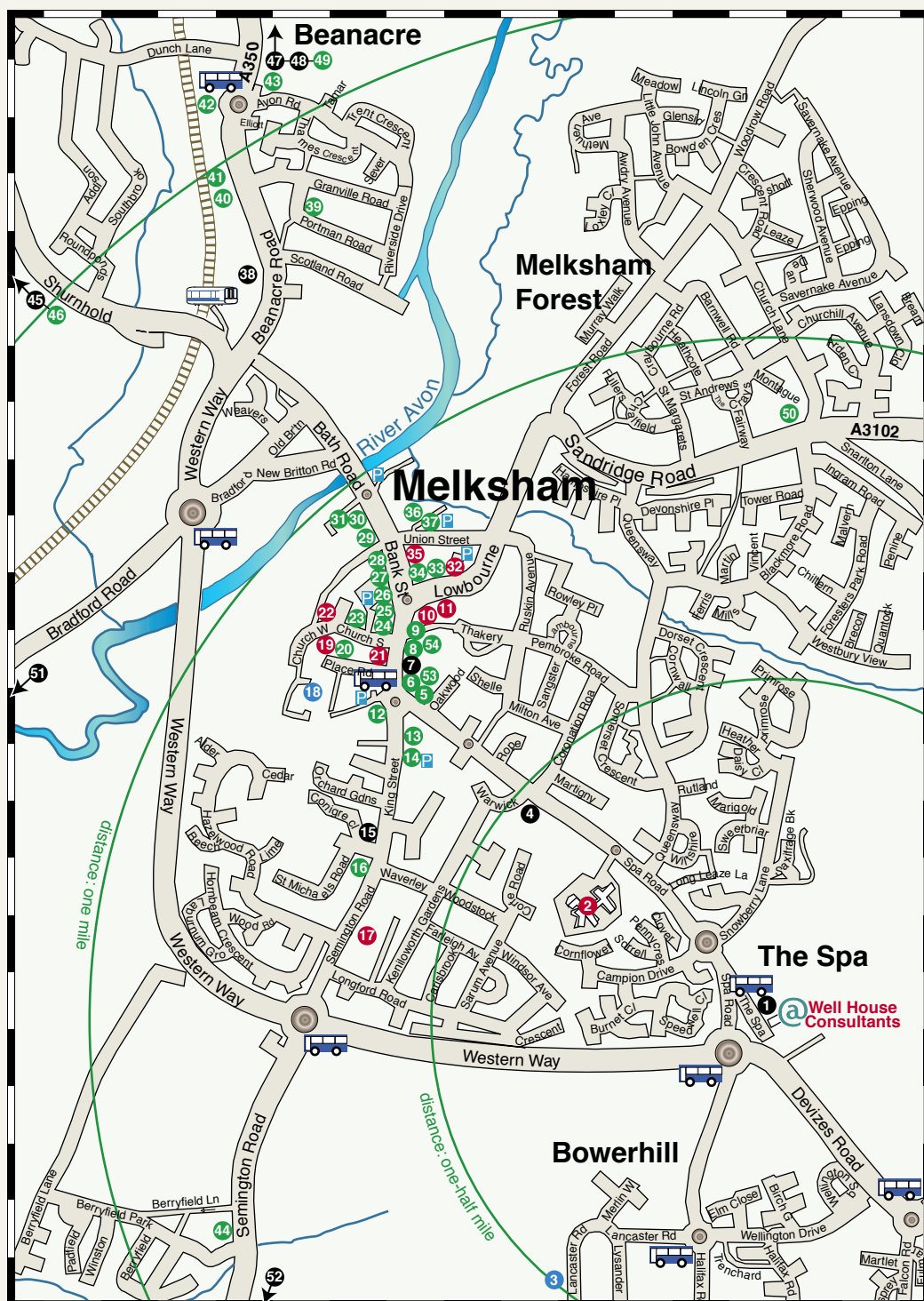
- 47 Beechfield House
- 15 Conigre Farm
- 7 The Kings Arms Hotel
- 38 Longhope Guest House
- 4 The Old Manor
- 45 The Shaw Country Hotel
- 1 The Spa B&B
- 48 Travel Inn
- 52 Travelodge
- 51 Woolley Grange Hotel



From the city or town to Well House Consultants' door
step as the crow flies* ... (in order of distance away)

Bath	10	Leicester	120
Swindon	20	Cambridge	140
Salisbury	20	Sheffield	160
Bristol	25	Manchester	165
Southampton	45	Liverpool	165
Cardiff	50	Dover	175
Oxford	55	Leeds	185
Reading	60	Norwich	200
Portsmouth	70	York	205
Swansea	90	Newcastle	275
Birmingham	90	Glasgow	355
London	105	Edinburgh	355
Plymouth	115		

* These are the mileages we use to determine our expenses for on-site courses.



Darwin evolves you to Java higher use in his cookbook

by Graham Ellis, Well House Consultants Ltd.



Java Cookbook

Solutions and Examples
for Java Developers

By Ian Darwin

June 2001

0-596-00170-3

882 pages

\$44.95 US \$67.95 CA £31.95 UK

A full synopsis of this book can be found on our website, along with a link on how to purchase directly from amazon.co.uk.

Please visit:

<http://www.wellho.net/book/>

0-596-00170-3.html

"This book is idea for anyone who knows some Java and wants to learn more". So says the author in his preface, and he's right. There are other books out there to teach you the basics and syntax of Java, books that will teach you about Object Oriented design, and others that go in some detail about a specific use of Java, such as servlets or JSPs. But what about those who wish to take Java that extra step beyond?

Have you ever thought "surely there has to be an easy method to ..."? That's where the "Java Cookbook" comes in. Its 800-plus pages are packed with information, all arranged in a problem / solution / discussion / see-also format which will rapidly get you on the right track when you make new uses of your Java knowledge in areas as wide ranging as string handling to working with graphics, and from Internationalisation to JSPs.

Ian Darwin has put a huge effort into researching answers to the questions that stump many newcomers to Java, and has catalogued his answers well. If you've already learnt Java, but don't yet know all the answers instantly, I would encourage you to add this book to your library.



The typist and the engineer

Do you need to test a new piece of software or hardware? You sit there, and you type in your tests, and you check the results. It gets much more difficult when you're testing something that's multi-user and interactive. Do the users interfere with each other? Does it work every time? Testing systems such as this can be time-consuming (and boring!), and it's easy to overlook errors if one set of answers is subtly different to another.



Expect (based on Tcl) allows you to automate testing such as this by using two commands – "send" and "expect" – to send a string out to the item under test, and check the response. Expect doesn't have a huge user base, but it's a life saver for those specialist requirements that need it.

No one can as type as fast as a computer can send information. Therefore, to make testing from Expect more realistic, an option to send (-h) lets the text be sent at "human" speed. You can even define the foibles of your human being if you wish, for example:

```
set send_human
{.1 .25 2 .05 1.5}
send -h "This is a message
coming out as if it was
typed by a typist\n"
```

In this example, there's an average gap of 0.1 seconds between characters in words, and 0.25 seconds between characters at the word end. The figure "2" defines the "shape", which is a measure on consistency; 2 is reasonable for a typist, a lower figure indicating a much more erratic person entering data and a higher figure giving greater consistency. The absolute minimum time between characters is 0.05 seconds, and the absolute maximum is 1.5 seconds.

Can you work out what the parameters mean for our second example, the engineer?

```
set send_human
{.3 .4 0.6 .1 3}
send -h "This is a lett\b\b\b\b
message coming out as if it was typed by
an engineer\n"
```



Cat or Dog?

In our last newsletter, we asked you to vote "Cat" or "Dog" on our website using a program written to demonstrate cookies in PHP4.

Charlie (that's her above) would like to thank the 16 of you that voted "Cat", but she does understand that there are some nice dogs around too, such as her friend Bertie.



(Thankfully
you decided
for us!)

T R A I N I N G T I M E S		
Let us know if you're arriving by train for your course, and we'll collect you from the station. Courses start at 09:45 on the first day, intentionally timed to coincide with the local train service. We usually train until 5pm, but we finish a few minutes early on the last	London (Paddington) Reading Swindon Chippenham Melksham	d. 07:45 (Change at Swindon) d. 08:10 (Change at Swindon) d. 08:46 d. 09:02 a. 09:11
day so that you can leave on the 17:02 train. You can also connect into the 09:11 arrival from Oxford at 07:45, Didcot at 08:09, Bristol at 08:15 and Bath at 08:38.		

What is the latest open source software version?

Table and comments were compiled in January, 2002

In this regular feature, we give you the latest news headlines from the Open Source world, and keep you updated on the latest releases of the core software and languages that we cover.

COMMENTS ON

Perl

The current stable version of Perl is 5.6.1, and the current "bleeding edge" development version is 5.7.2.

There's a Perl 5.8 in the works, consolidating and stabilising all the work done on 5.7.

Looking further ahead, development work proceeds on the internals of Perl 6 – see the article about parrot beginning on page 8 in this newsletter. Perl 6 has major changes and was commented on in detail in our last newsletter. Please ask us for a back issue if you missed it.

If you're happily running with Perl 5.005 or 5.6.0, you probably won't want to upgrade to 5.6.1. Perl has been around long enough to have included everything most people want for ages.

If you're running an earlier Perl, it probably is worth moving up to 5.6.1. Modules added at recent releases have increased functionality, and many of the CPAN modules that you may want to make use of require a reasonably recent version.

COMMENTS ON

Java

Stable versions 1.3.1 are now available for both the standard and the enterprise editions of Java.

There is a release 1.4 of the standard edition that is labelled as being a "release candidate".

In order to simplify the setting up of a web server using Java, a Web Service Developers Pack is now available for download from java.sun.com.

It includes Tomcat, Ant, the XML pack release, the JSP Standard tag library and the WSDP registry server.

COMMENTS ON

PHP

The popularity of PHP continues to rocket. Most users have now moved on from PHP 3 to PHP 4, with a phenomenal increase in performance achieved by the Zend engine.

There were a few very specific compatibility issues between PHP 3 and PHP 4, so do check your code before you upgrade your production

Subject	Stable Version	Development/Future	Web Site
Perl	5.6.1	5.7.2 (dev) 6.0 (future)	www.perl.com
Tcl/Tk	8.3.4.2		www.activestate.com
Expect	5.34		expect.nist.gov
Python	2.1.2	2.0.2	www.python.org
Java (see comments)	J2EE 1.3.1 J2SE 1.3.1	J2SE 1.4 Release Candidate	java.sun.com www.javasoft.com
PHP	4.1.1		www.php.net
MySQL	3.23.47	4.0.1 (Alpha)	www.mysql.com
C (Gnu)	gcc-3.0.3	3.1 (future)	www.gnu.org

system just in case!

The current issue of PHP is 4.1.1 (see separate article on page 5 in this newsletter).

The big change in 4.1.x from 4.0.6 relates to security issues of variables set up from forms, cookies and the environment.

COMMENTS ON

Tcl / Tk

Release 8.3.4.2 of ActiveState's build includes the latest Tcl, plus the following extensions:

- Tk 8.3.4
- [incr Tcl/Tk] 3.2 and iwidgets 4.0.0
- Tcllib 1.2 and Bwidgets 1.4.0
- tktable 2.7
- tkcon 2.3
- TclX 8.3
- expect 5.34 (Unix only)
- lmg 1.2.4
- TclXML 2.0 and TclDOM 2.0
- TclSOAP 1.6
- Snack 2.1.3
- TkHTML 2.0 (Unix only)

If you have a version of Tcl prior to 8.2, it's probably worth your while upgrading to 8.3.4.2, but if you're a user of 8.2 or an earlier 8.3, it's probably only worth upgrading if you want get the extra extensions in this build, or if you're looking for some of the specific changes made, such as improved efficiency in string split, or support for Unicode fonts in X.

COMMENTS ON

Python

Final Python 2.2 was released late in December 2001.

Python 2.2 is a cleanup of some of the darker corners of the language, and for cautious types there's also a 2.1.2 release available.

With the cleanup come some new capabilities in Python 2.2.

You can now subclass built-in types. You can also define static and class methods in addition to instance methods which were previously available.

Using properties, accessing attributes of an object makes for cleaner, easier coding.

For the moment, Python supports two styles of classes, the old style and the new style.

It is anticipated that the old style will be discarded fairly rapidly when there should be no compatibility issues involved. This may happen with the release of Python 3.0.

COMMENTS ON

MySQL

There's a Beta release of MySQL (4.0.1) out for testing and development work.

To some extent this is an "enabling" release which restructures some of the code internals to allow for further enhancement.

MySQL has been referred to as a fast and reliable relational database engine, but with more limited facilities than other engines.

The enhancements in 4.0.1 will help

redress that balance.

For production use, and for development where you are not pressing the bounds of MySQL, you might like to stick with 3.23.47 (or another recent version) just for the moment.

Remember, if you download the source code you can compile up MySQL to include or exclude support for database table formats such as Berkeley and INNODB.

If you download a pre-compiled binary, you would be able to choose between the standard version, which does not include many of the compile-optional features in MySQL, or the MAX version which does include most of them.

COMMENTS ON

C

Gcc 3.0.3 was released in December 2001.

Enhancements to Gcc over release 2.95 include many optimisations, and a number of additional ports.

Although originally the Gnu C Compiler, these days Gcc stands for the "Gnu Compiler Collection" and is capable of compiling a range of languages including C, C++, Java and Fortran.

The Java compiler is of particular interest because it can generate both class file and native code.

The native code libraries now include most of the common non-GUI classes.

Many improvements have been made in this area with the release of 3.0.3 of Gcc.

Keeping data secure as it passes through the Internet

When a visitor comes to a web site, the client process on his computer (the browser) usually calls for a particular item from the web site server computer using a protocol known as http (hypertext transfer protocol).

Http is a poor protocol for secret or personal data, since the information contained in the messages can be easily read and copied at any of the intermediate points on its journey from server to client and vice versa. And neither the client nor the server will know it's been copied.

Where information needs to be securely transferred between a server and a client, a different protocol is used, usually HTTPS. Using HTTPS, a browser will call for particular information via HTTPS protocol, and the server will respond also in HTTPS.

HTTPS information is scrambled ("encoded") by the sender in such a way that any copies grabbed along its journey will not be meaningful; only the recipient will have the necessary key to decode the information it contains. Thus https provides a data "tunnel" between client and server.

What is a secure server?

The words "secure server" are used to describe a server which can read requests in https, interpret them, and respond also in https.

Caution: Although the https incoming and outgoing are secured, you should note that saying that a computer is a "secure server" doesn't mean that all transmissions have to be made using the secure protocol, nor that information received via the secure link cannot be subsequently stored or retransmitted by that computer in other less secure ways.

What do I need to do to run a secure server?

1. A piece of web server software that includes the code to encode, decode, run https protocols and encode and decode using an "ssl" (secure socket layer) when required.
2. A certificate which the server can present, electronically signed by an official signatory, to confirm that the server really is who it says it is.

You'll have to be a big company to run your own web server (otherwise you'll share a computer at an ISP), and if you want such a server to run a secure service you'll need to get:

- a) A Web server that includes https

support, configured to have that support running. The Apache Web server, which is available free via the open source license and accounts for the majority of web servers, has this capability.

- b) A certificate; there are only a couple of organisations that can provide these, and most are provided by a company called Thawte Consulting.

See <http://www.thawte.com>

Can my

https

Level 4: Support for https on their web server computers, and a certificate obtained for each domain that's hosted there are required to use it.

Level 4 is a full service; chances are you will need to make special arrangements if you want this, as the cost of a certificate for each domain hosted on a server is significant and only a small proportion of domains actually need the service.

URLs written for the domain name of the host, rather than the domain name of your site. This is not a problem for intermediate pages in a transaction, and indeed it may re-assure site users when they see a separate host name being used for secure pages.

Level 2 provides the web site owner with some https services, but as the http and https are running on different computers it means that facilities to link data entered onto secure and insecure forms is very limited. For example, a credit card number entered onto a secure form wouldn't be available with the list of the parts the customer ordered from you on the normal server. The best you could do would be to get one set of data off each server when you came to fill the order, and then link up that data at the order filling location. Level 2 may also appear incomplete to the customer as the form that the secure server presents will not contain full details of his order; at best it could include an invoice total supplied to the secure server indirectly via a cookie or hidden field. It's good idea to use https to send out this form, so that he know he's been put to a secure service before he enters card details.

Level 1 support does not include https support. You could still ask your user for card details, but users would be reluctant to fill in your form and your credit card company and trade standards association wouldn't be best pleased.

What else do I need to handle secure data on my web site?

"Secure server" is a misnomer – it's a check that information is secured between the browser and the server, and says nothing about the security of the server computer.

Web site authors need to ensure the security of trusted data that they receive via https. For example, if the trusted data is a credit card number it should never be forwarded in a plain text email from the web server to the order-filling location. Care also needs to be taken that data stored on the server can't get into the wrong hands, especially if the server is shared.



ISP provide a secure server?

Yes, they can, although it's their commercial decision whether they do so and how complete their service is.

They could provide:

Level 1: No support for https

Level 2: Support for https and a certificate on one of their computers, allowing customers limited access to that computer.

Level 3: Support for https (and a certificate) on each of their web server computers, allowing visitors to websites hosted on that computer access via

Level 3

costs much less for the ISP to provide, as there's just one certificate-per-host

computer, not one per domain, and it's quite typical for a single host computer to service 500 domains. If you have a Level 3 service, your visitors will be able to use both http and https to and from your web site, place your own scripts there, etc. The downside is that because you're using the certificate for the computer as a whole, you'll need to have your https

Here's an example of an http and then an https request to a server at an ISP that provides support at level 3:

```
http://www.wellho.info/cgi-bin/submit.pl
https://cedar.he.net/cgi-bin/suid/~wellinfo/paydets.pl
```


The following public courses are scheduled at our Melksham, Wiltshire training centre. Dates listed are until August 2002; see <http://www.wellho.net> for further dates. If you have four or more trainees, we'll run an extra private course for you at our training centre or at your office, and tailored to meet individual company needs.

PERL

Perl is a programming language used in data manipulation, web interaction and systems administration. Its wide range of facilities make it ideal for large and small jobs, and it can act as an excellent glueware between otherwise unconnectable applications.

We offer two different foundation courses depending on your background, and two advanced courses to meet specific application needs.

Learning to Program in Perl

A foundation course for those who haven't programmed before, or who are rusty. Covers the fundamentals of programming in Perl, and the major features of the language that are required in most applications.

☒ 5-day course, runs:

Monday 20 May 2002 to Friday 24 May 2002

Monday 5 August 2002 to Friday 9 August 2002

Perl Programming

For those with programming experience. Covers the fundamentals of Perl by comparing the language to other languages, then proceeds on to the major features of the language that are used in most applications of Perl.

☒ 4-day course, runs:

Monday 4 March 2002 to Thursday 7 March 2002

Tuesday 7 May 2002 to Friday 10 May 2002

Monday 1 July 2002 to Thursday 4 July 2002

Using Perl on the Web

Using Perl on a web server through CGI, and other web-based interfaces such as automated FTP, clients, servers, modPerl, etc. An advanced course that assumes fundamental Perl knowledge.

☒ 2-day course, runs:

Scheduled Courses

Prices

1-day courses	£ 290.00	470.00 euros
2-day courses	£ 490.00	800.00 euros
3-day courses	£ 690.00	1,130.00 euros
4-day courses	£ 890.00	1,460.00 euros
5-day courses	£ 1,090.00	1,790.00 euros

For multiple students from the same company booked on the same course, simply calculate the total cost using the formula:

£200.00 for each student per day plus £90.00

(e.g. 2 students / 4 days = £1,690.00)

Prices are valid for all courses offered in this schedule and on our web site <http://www.wellho.net> for bookings confirmed by 31st August 2002.

Monday 8 April 2002 and Tuesday 9 April 2002

Monday 16 September 2002 and Tuesday 17 September 2002

Perl for Larger Projects

This advanced course covers topics that you'll need to cover if you're using Perl on a larger project. Content includes variable scoping, namespaces, object orientation, POD, and much more.

☒ 3-day course, runs:

Wednesday 10 April 2002 to Friday 12 April 2002

Wednesday 18 September 2002 to Friday 20 September 2002

Wednesday 18 September 2002 to Friday 20 September 2002

Perl Extra

Perl Extra is only available to individual trainees who have attended one of our public Perl courses, and require further advanced training in a combination not offered by our other advanced training.

☒ Dates are scheduled to be mutually convenient to the trainee and the trainer.

JAVA2

Java 2 is a wide-ranging, portable, network-aware language that you may wish to run on servers, on clients, or on stand-alone hardware. We'll have you writing JSP pages, servlets and applets by the end of your training, no matter whether or not you've programmed before.

Java Programming for the Web

This course takes the newcomer to programming right through to writing JSP, servlets and applets. It covers the world of Java, programming in Java, and the programming interfaces used on Java web-based applications.

☒ 5-day course, runs:

Monday 11 March 2002 to Friday 15 March 2002

Monday 10 June 2002 to Friday 14 June 2002

Java Extra

Java Extra is only available to individual trainees who have attended our public Java courses, and require further advanced training.

☒ Dates scheduled to be mutually convenient to the trainee and trainer.

PHP

PHP is a language that you embed in web pages to run on your server. You should already know something of the web, HTML, SQL and programming. We offer two courses: a technology day that covers enough of those prerequisites to get you started, and a PHP programming course that covers both the language and its web use.

Technology for PHP

The structure of the web, web pages through HTML and relational database access through SQL covering the prerequisites needed to follow on to our PHP programming course.

☒ 1-day course, runs:

Monday 25 February 2002

Tuesday 28 May 2002

Tuesday 27 August 2002

PHP Programming

This course covers programming in the PHP language, and also topics such as state, cookies, interfacing to SQL, and emailing from the server that you'll need to write effective server side code.

☒ 3-day course, runs:

Wednesday 29 May 2002 to Friday 31 May 2002

Wednesday 28 August 2002 to Friday 30 August 2002

MYSQL

This course introduces you to the world of SQL database engines. We study the SQL language and how it's used from PHP, Perl and Java, and administration including user account and security controls,

and database design so that you'll be putting together well thought out, maintainable, expandable databases.

MySQL

Our MySQL course covers the structure of a relational database, the SQL language, database design, installation and administration of MySQL, and its use from open source languages.

☒ 2-day course, runs:

Monday 29 April 2002 and Tuesday 30 April 2002

Monday 23 September 2002 and Tuesday 25 September 2002

TCL, TCL/TK and EXPECT

The Tcl language is a scripting language with a wide range of extensions, such as Expect and Tk, which make it very effective for applications such as writing Graphic User Interfaces and automating repetitive operator tasks.

Tcl and Expect Basics

This course teaches you Tcl, from fundamentals through most of the major features, and also the Expect extension, but not the Tk extension.

☒ 3-day course, runs:

Monday 18 March 2002 to Wednesday 20 March 2002

Monday 22 July 2002 to Wednesday 24 July 2002

Monday 22 July 2002 to Wednesday 24 July 2002

Monday 22 July 2002 to Wednesday 24 July 2002

Tcl/Tk and Expect

This course covers the Tk extension to Tcl which provides the Graphic User Interface. Learn how to build widgets, how to pack them into frames, and how to use them to provide interactive, GUI-based programs. A knowledge of Tcl is a prerequisite of this course.

☒ 2-day course, runs:

Thursday 21 March 2002 and Friday 22 March 2002

Thursday 25 July 2002 and Friday 26 July 2002

We have noticed a move by other training companies in the past few months to ask for 50% (or even full) payment with order, and we ask "what are they afraid of?". Are they afraid that their customers might withhold payment because of the poor quality of their product?

Are they in financial trouble, and if so, are you sure they'll be able to run that course?

When you book a course with us, we ask that you confirm your booking with an official order, or guarantee the booking with a credit card. We won't ask for any

payment until you actually attend the course. Perhaps we're old-fashioned, but we don't believe that you should pay for goods until you have received them.

We are also aware that a number of training companies are fighting for survival after the Department

for Education and Skills (DfES) suspended the Government's Individual Learning Account (ILA) programme. We did not rely on this scheme and will continue to provide high quality training with unchanged terms and conditions into the foreseeable future.

Learning comes from life experience, a bit of planning

by Graham Ellis, Well House Consultants Ltd.

It's hard to find the right IT staff to fill your vacancies and to retain the good staff you have already. That's in spite of the fact that we're in something of an economic downturn at the present time, when you would expect there to be plenty of good applicants for each job, and a reluctance of staff to move on. Why this apparent paradox? Much of it can be blamed on the lack of adequate training over many years.

Education systems, which evolved over centuries, supposedly teach children and not-quite-adults the skills they need for their working lives. Those skills have been traditionally topped up by apprenticeships and other vocational training offered by firms to their younger employees, secure in the knowledge that they'll have many years of service from those employees to recoup their investments.

I left school in 1971, having just sat four A levels, on the eve of my 17th birthday. We hadn't had a computer at school, but I enjoyed maths and sorting out problems of logic, and thought that a career in computing would be interesting.

I took a job acting as tape librarian for a company that had computers with 32k of memory, were fed with punched cards, and had an electricity bill of £1,200.00 per month. I was very fortunate in that job; I was a sponge for information, and I learnt a lot, my employers provided a torrent from the

tap of knowledge.

A training for life, then? No, certainly not. Had I not gone on to other skill acquisitions over the years, then

really aren't helped by their employers.

It's estimated that more than 40% of employers of IT staff have no formal training plan, and it's not a priority item

vibrant environment where that can keep up to date they'll be much better motivated, they'll be more likely to be more productive, and they'll think twice

Internet users (000)				Number of Internet users – November 2001			
France	UK	Germany	Spain	Denmark	Sweden	Norway	USA
10 341	14 730	16 121	3 784	2 436	3 612	1 705	82 305
Profile of Internet users							Men / Women
Country			Men			Women	
France			57.1%			42.9%	
UK			57.5%			42.5%	
Germany			59.7%			40.3%	
Spain			62.5%			37.5%	
Denmark			57.9%			42.1%	
Sweden			57.4%			42.6%	
Norway			54.7%			45.3%	
USA			52.8%			47.2%	
Connection frequency				Days connected during the month of November 2001			
France	UK	Germany	Spain	Denmark	Sweden	Norway	USA
10.8	9.9	10.9	11.0	11.3	10.1	9.7	11.7
Days connected (per month)							Men / Women
Country			Men			Women	
France			10.8			8.4	
UK			9.7			8.0	
Germany			10.9			8.9	
Spain			11.2			8.8	
Denmark			10.6			8.4	
Norway			9.5			7.5	

Source: NetValue

today I would be like 47% of current information technology staff who are reported not to be fully proficient because of a lack of new technologies skills or experience. And these staff

for many other companies.

I could show you my c.v., a long list of what and where I've learnt since I left school ranging from the degree course that I went back to take, through operating system and product-specific formal training courses, conferences and meetings to reading books and other illustrations of "learning on the job". I shouldn't forget personal skill training either.

I still learn and I still want to learn; it's a sad day when I learn nothing. I could list some things I learnt about Java yesterday, and some Perl topics I learnt this morning. I'm lucky; I'm self-motivated to learn, and my life style and circumstances (my wife understands, for instance) give me the opportunity and encouragement.

Why do I tell you my story? I would not be where I am today if it wasn't for all that learning.

If you're an employer, look to providing training opportunities for your staff. You might think that they know their jobs (and perhaps they do), but do they know about emerging technical subjects that will let them do their jobs even better? Are you worried that you'll lose them when they've got those extra skills? You might ... but if you provide a forward-looking,

before they jump ship to take any offer that's tempted their way!

If you're a geek or a geek wannabe, go for it! The traditional education system has stepped forward over the years, but still doesn't provide adequate top-up and retraining, perhaps because there aren't the self-motivated takers for it.

Don't just rush into a commercial course, either, but come up with a personal training plan. Look ahead, see where you want to be in a year's time, invest your time and money wisely.

An opportunity came for me to attend Perl Whirl, a conference with Larry Wall, Tom Christiansen, Randal Schwartz and many other top industry names amongst the guest speakers and teachers. The cost? You don't want to know. The experience? Invaluable.

There are many ways you can learn that don't cost as much. Come on a commercial course, teach yourself through self-study by buying books or CDs, learn on the job.

If you're a bit like me, you'll find a tremendous satisfaction in using the new technologies, and you'll find yourself in a position as a valued, well paid member of a team.

Your contacts at Well House Consultants:

Administration: Lisa Ellis lisa@wellho.net
 Technical: Graham Ellis graham@wellho.net
 Phone: 01225 708 225 +44 1225 708 225
 (International)
 Fax: 01225 707 126 +44 1225 707 126
 (International)
 Web Site: <http://www.wellho.net>
 Postal Address: 404, The Spa, Melksham, Wiltshire, UK SN12 6QL

Our office hours are 08:00 to 18:00 Monday to Friday.

You'll usually reach us at weekend and in the evenings too, as we often divert the phone when we're not around. If you email us, we'll receive your email within 24 hours, 365 days a year.

This newsletter is published by Well House Consultants and distributed free. Please contact us if you're reading someone else's copy and would like to be added to the circulation list, or if you would like to be removed from our list.

All material in this newsletter is copyright Well House Consultants, Ltd., and may not be reproduced in any form or by any means electronic without prior written permission of Well House Consultants except in the case of brief quotations embodied in critical articles and reviews.

Well House Consultants has attempted throughout this publication



to distinguish proprietary trademarks from descriptive terms by following the capitalisation style used by the manufacturer.