

IAPetus



Quarterly Bulletin For the Institution of Analysts & Programmers

Issue 16 December 1995

Council Elections *Nominations for election in 1996*

The constitution of the IAP provides for a governing Council of fifteen elected members. Each member serves for 3 years, five members retiring and five new members being elected every year.

This year, the five people who were our very first Council members retire (although most will be standing for re-election) and – as they were originally appointed – when the 1996 election is complete we shall have our first totally elected Council.

We are now calling for nominations for the five Council places which become available on 1st June 1996. Only Companions, Fellows

or Members of the Institution can stand for the Council, but all members including Associate Members are entitled to make nominations and to vote in the elections.

Each member can nominate up to five people, because there are five vacancies to be filled. Nominations should be addressed to the Director General of the Institution, saying who you are and who you wish to nominate. Nominations received after 31st January 1996 will be ignored.

Those thinking of standing for the Council should know that Section 4 of the Institution's Memorandum of Association states "... no member of

its Council of Management or Governing Body shall be appointed to any office of the Institution paid by salary or fees or receive any remuneration or other benefit or money's worth from the Institution".

Members are expected to attend full meetings of the Council three times a year. The Institution does not normally pay expenses, but considerable effort is made to minimise the cost to Members. Meetings are held in varying locations to suit the convenience of those Members attending, and wherever possible the Council's business is conducted by letter, telephone or fax.

IAP Subscriptions Tidings of Joy from the Tax Man

We have just heard that the Institution has been approved by the Board of the Inland Revenue under Section 201 of the Income and Corporation Taxes Act 1988, with effect from 6th April 1995.

This is the result of heroic efforts by our Accountants, and particularly their Partner, Chris Fassam.

When the IAP was incorporated in 1993, the new Constitution was drafted with one eye on the 1998 Act.

As a result it is now accepted that the Institution's activities fall mainly within the requirements of

Section 201(3). Thus the Institution's name will appear in the next edition of the list of approved bodies which is due for publication in 1996.

Inspectors of Taxes will not receive notification of the Institution's approved status until then.

In the meantime members wishing to obtain relief on their subscriptions will need to explain to their local Tax Inspector that the Institution has only recently been approved, and quote the reference number SAPP/T1644/37/1984/JEM.

Standing Orders – still trundling in!

Some of our older members are paying the IAP more than they need!

A trickle of money is still being received at the Institution's obsolete Natwest account in Twickenham. Most of these payments are from existing IAP members who failed to cancel their standing orders when they changed over to paying by direct debit in 1992.

Please check if this might apply to you, and if so write to your bank telling them to stop the payments. The Institution cannot do this – your bank will only take instructions from you. But make sure they stop the standing order, *not* the direct debit!

The nights are drawing in, and Easter eggs are on sale in the shops – Christmas must be approaching!

The last *IAPetus* of the year; good resolutions disappearing under lashings of Christmas pud; office parties; frosty nights (well, there was an excellent meteor shower at 4 am on 18th November, but I doubt many of you saw that!)... ugh I hate winter.

What do you want to see in 1996? And what are you going to do about it? How about writing an article for

IAPetus, or standing for the Council. Maybe we'll see you on the Internet – the Institution will have its own presence soon.

The Council has several ideas in the pipeline on a range of topics – just watch this space! – and if there is something that you think ought to be considered, write or call and tell us about it.

This is YOUR Institution, be proud of it and be active in it. We ought to have an assortment of items

bearing the new Arms on sale soon, so buy them and wear or use them. If you want to use the Arms on your own letterhead, you can get a copy on disk from the Office (or by email if you ask me nicely!). As for being active... this is a "people" organisation, and if nobody does anything nothing happens. Get doing!

Happy Christmas, and may next year be even better than this one!

Megan C. Robertson

Software Failures

In the end, most software project failures come down to politics. We grind through endless methodologies to prove we are doing everything correctly, but no methodology yet devised supplies common sense where it is lacking in the people, and the conclusion of the report is never one the boss will disagree with.

In theory, everything we do is rigorously cost-justified. In theory, where I first worked, every program was written using Jackson Structured Programming Everyone was taught the basics, and the Job Description said it would always be used, on pain of death or worse.

In practice, nobody used it, and nobody was expected to except at the Annual Review. In practice, the project is justified by a conversation with the Chairman on the golf course, or to keep a senior manager quiet.

The few times we do try to cost justify some work, we can't. There are lots of reasons why. You can know something will save the company money, but proving it to the accountants who really run the joint is another matter. A realistic estimate of the cost is unacceptable, so an acceptable estimate is given, pared down, and accepted. There is never enough money to do it right, but always enough money to fix it once it is done.

That is why 95% of software projects come in after time and over budget. They wouldn't get started any other way. We think of that as a 5% success rate, but were those 5% of projects really successes? Do millions of peoples' lives really benefit as a result of them? Or were they the projects nobody cared about enough even to change the spec?

Some of that 95% are awful disasters, but some of them are the real success stories of our industry: projects that through ingenuity and imagination have improved the world people live in.

Projects that took about as long and cost about as much as the

people actually involved always knew they would.

What does it matter what the plans said if, at the end of the day, it actually works?

Paul Hazelden

New Fellows

We are particularly pleased to welcome the following new members who have been admitted to the Institution as Fellows since *IAPetus* was last published.

Jagath C.K. Basnayake BSc PgDip

A research officer in the Sri Lankan Department of Agriculture for 14 years, and Head of the Data Management Unit of the Central Agricultural Institute since 1988. Currently on secondment to the Sri Lankan Army HQ, responsible for all computer training and EDP.

Barry Compton Eleven years in

the RAF was followed by a further nine years designing and testing high definition colour scanners with Crosfield Electronics of Peterborough. This was followed by periods lecturing at Peterborough Regional College, as Computer Supervisor for the Royal National Institute for the Blind, and as a consultant to Ford.

John R. Dimmock Joined Heller-

man Deutsch as trainee on mainframes in 1980; left to go freelance in 1984. Since 1989 he has been a consultant with Link Associates, servicing a wide variety of clients includ-

ing Coca Cola, Schweppes, Smith Kline Beecham, Thompson Tour Operators and TNT Express Worldwide.

Michael L. Jackson BA IT

Manager for the Metropolitan Police, for whom he has worked in a variety of IT posts since leaving university in 1978.

Robert D. Linsdell Has worked in

software development since 1982, self employed since 1986. His earlier work on a variety of major assignments is dwarfed by his achievements since joining the JET Joint Undertaking in 1988, a project to prove the viability of nuclear fusion as a source of power. Initially Lead Commissioning Engineer, he now handles packages of work through his own company, Intelligent Microsystems.

David J. Whitmarsh A specialist

in the development of C tools. Ten years with Evolution Systems; Technical Director when he left in 1990. Project Manager with Medic Alert Foundation Europe until 1994. Currently on contract to Thomas Cook, developing a Back Office System where processing speed is critical.

The Director General writes

During the dark days of recession, a couple of years ago, my job (which is never easy, Council Members please note!) was made even harder by the succession of telephone calls I had to field from members who found themselves sadly out of work.

My advice at that time, basically to grit the teeth and press on, was probably not much help, and the very last thing they wanted to hear.

Now the situation is very different. I cannot recall when I last heard from a member who was short of work. On the contrary, Nicole and I are stretching our brains daily to locate programmers with the obscure skills needed to

satisfy a queue of potential clients. We have in fact returned to the situation which, until recently, was all that most members knew – a market with never enough programmers to satisfy the demand.

Let us hope this lasts for a long time. But let us also recognise that as the computer industry matures, as with all other industries, success is bound to lead to middle aged spread and a more pedestrian rate of growth. Inevitably talented people will migrate into an industry where salaries and prospects for advancement are good, but we will not always be able to rely on growth to absorb these new people. Increasingly extra people will mean smaller slices of pie

My aspiration is to ensure IAP members, at least, continue to feed well, and one of the Council's programmes which I expect to be revealing to you early in the New Year will address this point directly.

It is good to see so many members using the new Coat of Arms on personal letter headings. This helps the public recognise the IAP: it is good for us and for the individual members concerned. But we have now received recognition from another and altogether more exalted source – the Inland Revenue.

At long last the tax man has acknowledged the IAP as one of Britain's leading professional bodies, and our name has been added to that exclusive official list of bodies whose subscriptions qualify for automatic tax relief. (I did suggest this might be a good opportunity to put the subscriptions up again, but was howled down by a unanimous chorus of Council Members.)

During the year I and the other members of the Institution staff have received a great measure of support from the Council, and from very many individual members who have telephoned or written to us. We receive some marvellous letters; letters of encouragement, or effusive letters of thanks when we are hard pressed to think what we can possibly have said or done to warrant such a kind response. For that we thank you all.

1996 seems likely to be a prosperous year for programmers; with fingers crossed we should be all right at least until the General Election, and possibly beyond! My best wishes to you all for 1996, and a peaceful break over Christmas and the New Year.

Michael C. Ryan
Director General

Subscriptions for 1996

From 1st January 1996 the following rates will apply:-

Companions	£66.00
Fellows	£66.00
Members	£56.00
Associate Members	£46.00
Students	£36.00
Retired members	£36.00*

All these rates are reduced by £5.00 for members who pay by direct debit. For those with current accounts in the UK this is probably the cheapest and most convenient method.

Members preferring to pay by cheque or credit card will receive letters of reminder when their 1996 subscriptions are due. Subscriptions are payable on the anniversary of the day a member was first admitted to the Institution.

Overseas members may prefer to pay over the telephone by credit card (Access or Visa). This avoids the cost and inconvenience of purchasing sterling drafts.

* To qualify for the Retired rate a member must make application to the Institution, confirming that he is over 60 years of age, has been a member of the IAP for at least ten years, and is no longer working or available for work except on an occasional basis.

A Statement from the Director General of the IAP concerning the Institute of Computer Technology

Several members have written complaining about letters they have received soliciting business for the Institute of Computer Technology, which is obviously making use of our Institution's membership list.

We were not previously aware of the ICT, but it is listed in the London Telephone Directory at an address which is the business address of one of our Members.

The IAP membership list is only available to members of the Institution, who are bound not only by the IAP Code of Conduct, but by the conditions of supply which state that "No part of this publication may be reproduced, stored in a retrieval system or transmitted... without prior permission in writing of the Director General of the Institution of Analysts and Programmers".

The Council has accepted an explanation and an apology from the Member concerned, and no disciplinary action is proposed on this occasion.

Anyone can set up a professional body. As a result, while the computer industry is dominated by the BCS, with the IAP and IDPM in second place, there are also dozens of smaller organisations. The IAP has absolutely no objection to members joining other bodies: many IAP members are also in the BCS. It is IAP policy to promote good relations with the BCS and with all bona-fide organisations, but members must decide for themselves whether joining additional bodies will bring sufficient extra benefits to justify the extra subscriptions.

Mike Ryan

After three years of meeting in London, this year the Conference moved north to Birmingham (which is still south for me, but closer!).

Gathering in the Chamberlain Hotel, which we think started life as a workhouse, we were greeted by the Director General, Mike Ryan, who said that it was typical of a Treasurer to book us into a doss house! Nick Swain, the Institution's Treasurer, had organised the event although pressure of work meant that he wasn't able to be with us.

Methodology

The morning session was chaired by Council member Ted Pugh, who wasted no time in introducing the first speaker, Bob Hutchinson. Bob began by telling us that his background includes time as a politician, a preacher and a singer... but restrained himself to talk about how information systems are really about people and information flows; the computers are incidental.

Many "methodologies" are proposed by a series of gurus, but there is no substitute for a full and thorough understanding of the organisation, the people in it and the tasks that they want to undertake.

Too many people fall into the trap of thinking that if you select the appropriate methodology and use it correctly, this will result automatically in the production of the "correct" design for your information system. This sort of logic, if P then Q, cause and effect, is all right in science and mathematics but doesn't work too well in organisations. All a methodology is, is a formalisation of a natural process, and when it doesn't work, the answer is not to produce a more complicated one. Soft systems methodologies are just a more complicated way of making the same mistake!

The more that the system developer knows about the organisation and the technologies used therein, the better chance he has of producing a system which will serve that organisation's needs. The information system, to be successful, has to become an integral part of the organisation; and the development team should be regarded in the same light. Any organisation is dynamic and complex, and using a fixed methodology is rarely the best way to come in cold and analyse it.

People don't work like computers – they don't think in a binary manner, they use intuition and insight as well as data, calculation and analysis. The system has to reflect this, and to allow

for the needs of everybody in the organisation, not just those who will actually operate it. Others will, even if not directly, even if they never clap eyes on a computer, be involved in the information flows; and the system must cater for their needs as well.

Bob explained how he had, in 1991, begun to get to know the people in a freight forwarding company – not just the board members and the IT staff but everybody in the company. By 1994 he was ready to produce a workable IT strategy for the whole organisation. Now he could spot the correct sources and destinations of each item of information, and ensure that time was not wasted by, for example, people needing to re-key information that existed elsewhere.

The aim was to ensure that all members of the organisation were surrounded by information, and that systems were in place to enable them to extract that which they required or wanted to have. The starting point is that everything is available to anybody who chooses to access it – security measures only coming into play when absolutely necessary, rather than a "need to know" culture. To achieve this takes a long time, you can't do it by rushing in waving a methodology at the problem!

Machine errors

The next speaker was Simon Lewis, with the heretical idea that computers get their sums wrong! And worse, check procedures may give false confirmation of results. Simon explained that "machine errors", as he called them, applied only to mathematical operations, not to other forms of data handling, but even there they are too often ignored. People assume that computers are designed for doing mathematics, so won't get it wrong; they assume that because computers are digital they are flawless.

Machine errors are caused by limitations in the storage and handling of numbers; the most familiar case being rounding errors. As an example, Simon demonstrated "catastrophic cancellation", which can occur when the numbers being manipulated, say in the subtraction of one floating point number from another, are almost equal. The bit streams are the same in the high order bits, only the low order ones are different and so precision is reduced... the significant digits are in danger of being lost altogether.

So, you need to find "work arounds" that will avoid situations which may

give rise to such errors. Look for the likelihood of obtaining values which may cause problems, reduce the number of calculations done and use the most accurate rate

10 Conf

method that you can find to carry them out.

Don't split calculations, as internal precision can be lost during transitions... by now the non-mathematicians were longing for their coffins and the rest of us were looking for a pencil and a piece of paper to try out the examples for ourselves (my husband did when we got home that night!).

Personal tax

Coffee and tall tales later, we regathered to hear Neil Graham talk about the impending changes in personal and company taxation, in particular the introduction of self-assessment of personal tax.

This has been heralded by a series of cartoon adverts featuring a tax man who was to have been called "Hector the Inspector" until somebody thought that people might compare him to Hannibal Lector from the movie *The Silence of the Lambs* who had a distressing habit of biting chunks out of people... hmm, definitely sounds like a tax man to me!

Neil reckoned that the book-keeping required by self-assessment wasn't much more than is required of any small company; although he did warn us that Her Majesty's Customs and Excise could well be regarded as fiscal terrorists. Visions of a chap in bowler hat, pinstripes and Kalashnikov rifle!

Recruitment

The final speaker before lunch was Julia Robertson, who is the Chief Executive of the Institute of Employment Consultants. She'd come to present the case of the recruitment agent, to explain how professional they really are, and much maligned by many of those who have had cause to deal with them – particularly those of us who earn our bread in the contract market.

The IEC have been busy improving

the professionalism and competence of members, and Julia was full of useful tips for those who wish to use their services. Things like communica-

95 reference

tion, like checking on your consultant at least as thoroughly as they take up your references, work with them, choose a few people who really understand what you can do and want to do rather than spend your CV round as many agencies as you can find.

Wondering if we were wrong in considering all agents to be leeches and rip-off merchants, we staggered off to lunch. During which Julia asked me to write an article for their journal putting the computer professional's side of the story... perhaps this sort of communication, along with some of the things the Council have in store, will improve the relationship between our two professions.

Coat of arms

Well-fed and wine-d – although the tiled dining room made several people wonder if they were eating in a public convenience! – we returned to the conference room to participate in the Members' Forum. This was chaired by the Institution President, Jim Bates, who was attired in great splendour in the formal robes of a Fellow of the IAP.

I tried to retreat to my quiet corner at the back, but was winkled out to sit at the front with the rest of the Council members present... and then invited to talk about the new Coat of Arms. The grant of arms scroll had been brought up from the Institution offices, and this provided a good opportunity for everybody to see it in full mediaeval splendour.

Most of the discussion revolved around the need to raise the Institution's profile in the public eye, and how to ensure that the computing profession gains the recognition that is its due.

Product specification

Then John Weller talked about how to specify a fixed price contract. To set a fixed price, you need to fix the speci-

cation... which also helps avoid the "But you never said that you wanted THAT!" moment that all software developers dread.

John outlined the specification process that you need to follow; beginning with a feasibility study and producing a concept document and getting input from your customer long before the product specification itself is produced.

The concept document should include the understanding that you have reached about the nature of their business and its requirements in general, as well as the specifics of what you intend to do. Introducing these enables you to check with the customer that you have grasped what is going on, and lets them correct you if you have anything wrong before it is too late.

The product specification has to be extremely detailed, and should contain your description of the client's business, their requirements for the system you are to write for them, the environment (both software and hardware) in which it is to run and a full overview of the system itself. Nail everything down now, and get them to agree to it before you go any further. Detail the data structures and give screen descriptions; even specify the media on which the system will be delivered once you've finished writing it.

Outline your plans for security, both how you intend to ensure the safety of the data via a backup regimen (or state unequivocally that you aren't doing anything about it, but that they should!) and how you will provide access control to the application if this is deemed necessary.

Detail the test procedures to be followed. Of course, you'll make sure that the system goes before you deliver it, but exhaustive testing under full load is not usually possible except in active use... so do you want to propose a "You test and I'll fix" programme, and if so, what limits do you wish to impose? Specify the documentation that is to be provided, also the scope and duration of training.

And make sure that the really important bits are covered. Who will own the rights to the source code? What is the delivery and payment schedule? Use this full product specification as your contract with the customer, then nothing will be left to chance.

John recommends the use of a signature sheet – send the client two copies of the document, signed and do

nothing at all until one copy, countersigned by them, has been returned to you. It may be preferable to detail costs on a separate sheet – the document may be circulated amongst people who you (or the client) may not wish to know just how much it costs – but everything else ought to be in there. And just in case anything goes wrong, keep your milestone invoices under the limit of the Small Claims Court!

Litigation

Tea consumed, the final speaker for the day was Mike Cookham, who said that he was an ex-Lloyd's broker (and possibly the biggest liability they had!) and now dealt in matters of professional indemnity insurance as a specialist liability broker.

There is an increase in litigation against professionals of all trades, with a concomitant increase in the concept of professional indemnity. Many trades, for example accountancy, now hold it compulsory.

Still, avoid litigation as much as possible, but if someone sues you at least you will have the wherewithal to fight them and, if all goes pear-shaped, the ability to pay compensation without going broke!

And make sure that what you were supposed to have been doing has been clearly defined in the first place. It's all too easy to end up with the customer saying that you haven't done what you should have whilst you are certain that you have done all that you were hired to do!

The conference wound up, and the Annual General Meeting of the Institution was held – a brief event covering that which is required, such as the acceptance of the accounts. And so to the bar – I ended up discussing cognitive psychology over a pint! – and off home to our computers.

See you all next year!

Megan C. Robertson

IAPetus is the Quarterly Bulletin of the **Institution of Analysts and Programmers**. The Editor is Megan C. Robertson. All views expressed herein are those of the authors, and do not necessarily reflect the Institution's or *IAPetus'* opinions or position. All material is © Copyright The Institution of Analysts and Programmers 1995. Produced by Breeze Ltd, 0161-796 3600. Correspondence about *IAPetus*, contributions etc. should be sent to the Editor at 12 Bude Close, Crewe, Cheshire CW1 3XG (Tel: 01270 500565, email: mrobertson@cix.compulink.co.uk). Correspondence about the Institution should be sent to Charles House, 36 Culmington Road, London W13 9NH (Tel: 0181 567 2118, Fax: 0181 567 4379).

This article contains the distilled experience of a friend, not in the computer industry. Listen and learn from what he has to say.

Michael Ryan has asked me, not a member of your Institution, to say how I have gone about finding an initial client base and then reducing it to a "Front Line Listing" which can provide me with the workload that I require as a self employed consultant. Briefly, my background was a full-time career in the construction industry working through the Plant and Transport Divisions of a range of contracting companies with my last job as Director and General Manager of that division of Wimpey Construction Ltd.

Some two and a half years ago I was told by my superior that my skills and experience were no longer required by the company and I was given early retirement. I had always intended to practise as a self employed consultant after I had retired at the normal age but, of course, this situation had now been brought forward.

I had collected over a period of two to three years adverts, other information, etc., from various sources seeking experts in my particular field for short term assignments and I, therefore, used these as a starting point for marketing my services. There were only seven or eight such references.

It was clearly necessary to increase the number of organisations to which I needed to send my CV with a letter written to suit the actual addressee. I started by using directories published annually by the Institution of Civil Engineers and selected companies and organisations which I judged could make use of the services that I was offering. An important element of my approach was that I was willing to work overseas.

These included Building and Civil Engineering Contractors, Consulting Engineers with listings indicating that they could be involved with projects where my experience and skills could be used. Statutory and similar bodies, such as Crown Agents, ODA and Commonwealth Development Corporation, responsible for overseas aid projects. County Councils known to use their own resources in construction work and relatively close to my home address. Members of the Association of Management Consultants who have a department for con-

struction business and indicated that they could give work to me.

In each case I made the judgement whether a particular firm or organisation was worth a formal approach. This was based on my career experience and my own personal evaluation.

Focusing your client base

In as many instances as possible I telephoned the targeted firm and tried to talk to the person responsible for the type of work that I was seeking. In most cases the switchboard operators were very helpful.

Two words of warning – Personnel Managers are not much use when a specific project is still under development in that office and there are a lot of people who will assure you that they have responsibility for your area of specialist work but before long it becomes clear that this is not so. Patience and luck are indispensable.

This process of selecting targets and sending formal letters with full CV continued as a major part of my work programme for some eighteen months. However, almost more important than these two steps is the follow up with your contact names who are either known to you from your research or from the replies sent to you. It is paramount that you make and keep contact with that person. Do not be put off with non-availability. Keep phoning until you speak with them. Do not waste your call. Make a record of each call. I do this on my file copy of the relevant letter.

I called each client every two to three months for an initial period of some eighteen months and this procedure plus the assignments that I had carried out made it relatively easy to decide which client was serious and which was unlikely to provide me with work.

Towards the end of this period of eighteen months, during which I was fortunate to obtain assignments from each category of client, apart from Management Consultants, I gradually reduced my initial number of clients, 50–60, to between 20–25 who were serious in their ability to give me work or who had already done so.

This list includes firms and organisations from Europe and the USA as well as the UK. As my business is

now reasonably well established and mature I have extended my frequency of contacting those on my reduced list to around four months, fitting in with my work load. I am still making initial contacts where I believe there are real prospects of work and this process will, of course, continue for as long as I am in business.

Drawing on my own experience I would suggest the following points as very significant to anyone trying to develop a consultancy business where he or she is the sole consultant involved, certainly in the initial stages:

Produce and use a full up to date CV which does not undersell your experience and skills and includes a positive statement of what you offer to prospective clients.

Keep your CV up to date as you carry out work, two to three times a year if necessary.

Select as broad a base of initial clients to be approached as your judgement of their suitability to your business will allow.

Contact each of your initial targeted clients to locate the individual who should receive your letter and CV. Most adverts that bring positive results carry a name which should be used.

Approximately two weeks after sending letter plus CV follow up with a successful phone call to the named individual. This should establish a personal relationship between you and your client.

Repeat this telephone contact at suitable intervals of time, neither too long nor too short, for say up to fifteen months and if no work has come to you, you will need to make a decision whether you put the lead into a "semi dead" category or keep it as "first line". It is essential that you have a proper filing system for all your telephone calls and correspondence. If your contact has a secretary make sure she is on your side. This can be invaluable when you are trying to reach him with a routine call.

Develop a disciplined approach to the whole business of your marketing effort.

GOOD LUCK!

*R.H. Wykes BSc (Eng)
FICE FIMechE*

Keynote Concepts in Object orientation (wrt C++)

To crystallise my understanding of object-orientation I've tried to define the pivotal concepts, and essential terms, that characterise the 'new' discipline.

I thought that the Institution might like to publish the set in *IAPetus*, and bravely invite criticism – I'm sure there'll be plenty as it's a young subject with volatile frontiers.

The fundamental definitions are as generic as I can make them; however the *cognoscenti* will detect a C++ bias. For brevity, I've mostly omitted terms that are the common currency of modern computing.

Also, the definitions are sorted into topological rather than alphabetical order (intrinsic terms are not used until they have been defined) so the sequence can be read from beginning to end with optimum continuity.

alpha\beta: the first\second letter of the Greek alphabet (dummy entry to demonstrate usage of backslash notation in this article)

object orientation: paradigm shift from conventional free-standing functions to the encapsulation (latent association) of data and pertinent methods

[data] abstraction: the essence lifted clear of the detail – operational invariants that epitomise behaviour without being committed to an underlying type (such as a queue mechanism that can deploy entities of any single queueable type)

information hiding: data integrity regimen that characterises object-oriented systems by which information is accessible (on a need-to-know basis) via an invariant interface that conceals the data representation; thereby promoting reliability and potentiating reusability

scope: purview in which a distinct identifier denotes a particular variable (the same identifier is either unappointed in another scope or denotes a disparate variable in that scope)

object: that which has a boundary delimiting the space it occupies – nonprimitive computational objects encode the repertoire of their own behaviour and (ideally) retain an internal locus of control

object specialisation: object com-

monality engenders their classification into types, and any one type can be construed as a special case (subclass) of some more general type (superclass) so the ensemble forms a class hierarchy

inheritance: the propagation of attributes (data and method definitions) down a class hierarchy, from superclass to subclass, without explicit restatement of those attributes – class inheritance is typically augmented by new attributes and, unless restricted, these 'acquired' characteristics are heritable thereafter (this is a form of Lamarckian inheritance – nonDarwinian)

class: finite set of objects having identical behaviour or their progenitive abstraction – the basic unit of encapsulation; containing both function members (methods) and data members (variables) which can be individually scoped as private (concealed and uninheritable), protected (concealed but heritable), or public (accessible and heritable) – the class construct (aka multiprocedure module) originated in SIMULA67 and was later emulated by C++ and other OO languages

metaclass: finite set of classes that are not themselves metaclasses (restriction avoids Russell's paradox and similar absurdities)

instance: an object regarded as an entelechy of the class to which it belongs; with the class definition being the pattern for instantiation

persistent object: instance that outlasts (albeit dormant) an incarnation of the manipulating system – persistence is controversial, since purists contend that objects should be ephemeral, but it is often essential to pragmatic design

binding: symbol resolution by which identifiers are bound/mapped to the entities they denote (in respective scopes) and their types are rendered – early/static binding is done at compile time or link/load time, whereas late/dynamic binding is done at runtime (program start, procedure initiation, or statement execution time)

strongly\statically typed: pertaining to a language in which the type of every expression\variable can\ must be determined\ declared at compile time (Eiffel is strongly typed, C++ is statically typed)

dynamically typed: pertaining to a language in which types are not assigned in variable declarations and are therefore determined at runtime (Smalltalk is dynamically typed)

static class: declared object type, although the actual object is also permitted to be an instance of a derived class (in statically typed OO languages)

dynamic class: actual object type (the class of the value) as distinct from the declared type (the class of the variable) which could be ancestral

cardinality: any property possessed exclusively by all sets equipotent to the given set, where two sets are declared equipotent if there exists a one-to-one mapping of one set onto the other – elementary counting establishes such a mapping (bijection) to the positive integers, so the cardinal number of a class is the smallest ordinal number equipotent to that class (it represents the number of extant objects belonging to the class or the maximum number that can be instantiated)

sender\receiver: requesting\providing object in a message-mediated service transaction

message passing: object collaboration modality whereby a sender causes a receiver to activate a method

method: function member of a class, which is typically invoked in a message-passing style

this: pointer to self from inside a method, which upon dereferencing yields the value of the receiver for the message that caused the function to be invoked

static method: one having a unique procedure that is determinable in early binding and therefore obviates message passing

dynamic method: one having a pluripotent procedure which cannot be determined until late binding and is therefore activated by message passing

Continued on Page 8

Keynote concepts in Object Orientation – Continued from Page 7

method designator: nominal function identifier in message-passing expression that necessitates runtime selection (late binding) of the appropriate procedure

qualified name: method designation that indicates the class explicitly so activation by qualified name can be performed using procedure calling (early binding) rather than message passing (late binding)

constructor: optional static method, having the name of the class in which it is defined, that is automatically invoked to allocate and initialise objects of the namesake class (when they are defined)

destructor: optional static method, having the name of the matching constructor prefixed by a tilde, declared in the 'namesake' class whose objects it is automatically invoked to terminate (when they go out of scope, unless they are preserved)

ako relation: is-a-kind-of relation between a class and its superclass (eg primate is a kind of mammal)

is-a relation: condition in which any instance of a class is effectively a representative of an immediate superclass (eg man is a primate, so man is a mammal)

multiple inheritance: derivation of subclass from more than one immediate superclass – the dimension of the minimal inheritance list being at least two

private inheritance: derivation whereby the subclass exhibits the characteristics of an immediate superclass but instances of the subclass cannot be assigned to variables whose declared class is the superclass, and consequently the is-a relation is not preserved

inheritance graph: derivation schematic which shows the class hierarchy

reference variable: alias parameter (tacit pass-by-reference) which refers to another variable without explicit address-assignment or dereferencing

single-assignment: pertaining to 'variables' that are either syntactically immutable (const[ants] and reference variables) or semantically immutable (constructor-only assigned data members)

[procedural] overriding: subclass method taking precedence over identically-named superclass

method during the late binding required to resolve a message

[pure] virtual function: [deferred] generic method of superclass that can [and must] be overridden in the derivation of [instantiatable] subclasses

abstract [super]class: ancestral type without direct instances (containing at least one pure virtual function) thus serving exclusively as a base from which other classes inherit

specification class: abstract class that defines an interface which is to be implemented in descendant classes

rapid prototyping: exploratory programming technique involving the expeditious construction of pilot systems with which users experiment; and thereby generate feedback to enhance or retarget the main development – considered facile/natural in an OO/WIMP environment

friend: function/method/class explicitly granted access by a class [declaration] to otherwise private/protected members of that class

obligate members: methods that must be members such as constructors, destructors, and virtual functions

facultative members: methods that could otherwise be installed as friend functions/methods argument signature number and type of parameters in a particular call or message passing event

[parametric] overloading: association of more than one procedure with a single function/operator identifier (in the same scope) which is then disambiguated at runtime by inspecting the argument signature

I/O operators: extractor\inserter functions which perform input\output on all built-in types and can be further overloaded to encompass input\output on user-defined types

polymorphic entity: a variable that can assume different types, or a function having such an argument (pure polymorphism) or a function that has been overloaded (ad hoc polymorphism)

coupling: the kinds of connectedness between modules which, ranked from strongest/worst to weakest/best are, data coupling (direct access to data in an

instance of another class or reliance upon global data common to another class), sequence coupling (one class has a prescribed operational order but another class has control), parameter coupling (through argument signature and return type), and subclass coupling (objects [at least] behave like instances of their immediate superclass)

cohesion: the kinds of connectedness within modules which, ranked from weakest/worst to strongest/best are, random cohesion (unrelated methods arbitrarily grouped), utilitarian cohesion (functionally similar methods but no shared data or internal locus of control), temporal cohesion (time-clustered methods which are activated in short-order), communication cohesion (input-output associated methods that handle similar data or the same device), sequential cohesion (concomitant methods that must be activated in a prescribed order), functional cohesion (operationally allied methods that perform a circumscribed activity), and data cohesion (orchestrated methods that fully implement a complete data abstraction)

template: generic definition of a function or class (using formal type-parameters) that serves as a pattern by which type-specific working cases can be arbitrarily generated – generic functions are automatically self-overloading for unbounded operational polymorphism, whereas generic classes are parameterized types which also provide unbounded structural polymorphism – the latter facilitates, inter alia, generalized compound types such as container classes whose instances hold objects of another type (ie homogeneous aggregates exemplified by linked lists and associative arrays where the index need not be an integer)

Stephen Cumbers

Director of Apollo Data – a firm of independent computer consultants.

Tel / Fax +44 (0)1689 835094
email apollo@newdelphi.com...

Steve sent his article by email, which saved a lot of typing!!