As we mentioned earlier, an understanding of BASIC is not necessarily an open door to the computing industry. Although it is a popular language on home computers, most professionals regard it as badly structured and consider that it encourages bad programming habits and sloppy thinking. This is a real problem as most children with home computing experience are likely to have learned BASIC rather than one of the better structured languages such as LOGO or COMAL.

Several universities and colleges that run courses in computing now express a preference for entrants *who have not learned* BASIC as they feel that the language forms habits that are hard to break.

Despite this problem, many youngsters are finding a way to make their BASIC programming skills pay off. Many are writing games in BASIC that appeal to other youngsters, and software companies are anxious to get hold of games that are so directly targetted to an adolescent mind. Some of the 'whizz kids' featured in newspapers as earning vast sums at a young age write only in BASIC, and have no real understanding of computing. Others are genuine phenomena, writing in Assembly Language (the low-level language that controls a microprocessor's machine code very efficiently) and set for a very bright future indeed. Newspaper reporters are seldom qualified to distinguish between the two, and such reports can lead parents to think that their computer-obsessed offspring is ready to enter the big money. It is possible, but unlikely.

Inside The Business

In the computer industry proper, programmers are split into two groups: applications programmers and systems programmers. Applications programmers write programs to carry out a specific task. Systems programmers are 'housekeepers', writing programs to keep the computer system in order — to detect faults, for example. The applications programmer is likely to meet people outside the computer room — clients — and is likely to work as part of a team developing programs for a specific task. The systems programmers are more specialist, and tend to work alone. They are talking directly to the machine's 'intelligence'.

But at this point, the computer industry draws an artificial dividing line, beyond which it denies access to all but the brightest programmers and the best-qualified university graduates. This is the realm that belongs to the systems analysts and designers.

Systems analysts consider a problem and then decide how a computer can help to solve it. For example: an oil company discovers a new deposit under the sea bed. They have measured the extent of the deposit and found that the quality of oil varies widely. The oil company has to decide whether or not to invest the billions of dollars necessary to exploit the oil field. This decision will be based on projections about the state of the international oil market for the life of the field (say 20 years) and the company must decide which part of the field to drill first. Because the investment is so vast, the oil company hands the problem to its computer people for analysis. The analyst considers the problem, consults economists, oil marketing experts, geologists and other specialists and over a long period of time constructs a computer 'model' of the oil field.

The oil company executives can then play 'what if?' with this model, discovering how various decisions about price, refining techniques and market approaches would affect overall performance, and they are provided with all the information they need to make their final decisions about how best to exploit their field.

There are several other important roles in the computer industry, although few are as highly regarded as the systems analyst. Perhaps the exception is hardware design. There are openings for electronics engineers at all levels from high street repair centres to research departments, but the areas of product development and pure research are only open to those with the highest electronic engineering qualifications.

Many analysts and designers of both machines and software move on to managerial and consultancy positions, but these titles often indicate only that the individual is working in a more powerful role, very often self-employed. The work content of the job often remains the same.

On any typical day there is a massive shortfall in skilled computer personnel — some put it as high as 20,000 or more in this country alone — and at the same time a huge pool of unemployed, many of them graduates of universities and polytechnics. This obvious skills mis-match is a source of worry to educationalists and industrialists alike, and serious steps are being taken to rectify the situation, including re-training programmes for those qualified in other fields and a much wider variety of opportunities to learn at primary, secondary and tertiary levels.

Several governments, Britain's in particular, consider that microelectronics may provide an answer to some of the short-term unemployment problems. The Youth Training Scheme, which aims to provide 'on the job' training and work experience for unemployed school leavers, now offers 4,500 places at Information Technology Centres in Britain. At these centres, young school leavers learn about various aspects of microcomputing while receiving a training allowance equal to unemployment benefit. Other projects within the scheme offer some computer familiarisation to those who fell through the net at school (either because they left school before the computer arrived, or because they weren't 'selected' to use it) and also improve their prospects of finding a job, because for those who leave school without any computer familiarity or literacy, employment prospects can seem grim.

COURTESY OF THE SUNDAY TIMES

David Simmonds

David Simmonds, 17, earned himself £10,000 during his summer holidays.

He's a programming wizard who writes programs for Commodore (makers of the PET and Vic computers). Unlike many teenage boys, David writes 'serious' software that has commercial applications and he is expecting to find a lucrative niche in the computer industry when he has finished studying.

David started playing with a computer his father brought home from work, but he quickly abandoned game-playing and got down to discovering how to program. Initially David had some of his programs published in Commodore's user magazine and slowly but surely he began to sell copies of his programs, for a few pounds

Commodore eventually took notice and David persuaded them to let him show them what he could do. The result was his first serious programming assignment

Eugene Evans

Eugene Evans is 17 years old and his earnings are reported to be £40,000 a year!

Eugene is one of the many whizz kids now springing up in computer programming and he is helping to keep his employers, Imagine Software of Liverpool, among the top computer game producers in the country.

The high earnings made by these programming wizards usually take the form of royalties on the sales of games (rather like authors' book royalties) and teenage boys are best suited to develop games that will appeal to other teenage boys — the main market for computer games