

A GUIDE TO THE

INPOT

**COMPUTER
PROGRAMMING
COURSE**

INTRODUCING

INPUT

THE GREAT NEW WAY TO GET TO GRIPS WITH YOUR COMPUTER

Computers are one of the most powerful forces in the modern world – and they are becoming more influential all the time. And now that home computers are readily and cheaply available, there's plenty of opportunity for everyone to enter the fascinating world of computing.

- INPUT is the exciting new way to learn about computers. Learning to program a computer is a bit like learning to play football. You could spend ages reading all about the game first, but sooner or later you have to get out on the field and play. If you learn through actually doing it, it's likely to be far quicker and much more enjoyable, too. It's just like that with computer programming. As soon as you start to type in your first program, you begin to learn. With the right sort of programs, you'll have fun, too.

- With INPUT, you get straight into the world of computers. In each issue, you'll find the next part of a complete

and fascinating new programming course. With it, you'll be able to delve deeper and deeper into new and ever more exciting areas of making the computer work for you.

- The INPUT course concentrates on the aspects of programming that most people want to know — things like educational programs, data systems, record keeping, games programs and graphics. And it gives you an exciting new way to learn programming — written specially for your computer

Anyone can learn programming. You do not need to be an expert mathematician, although natural mathematical ability may well make you a better programmer. And, contrary to popular belief, there is NO age barrier.

- INPUT is designed to give you real computer knowledge without making it



boring. That's why the course is in easy stages, to take you forward at the pace that suits you.

Anyone can use a computer, but many users never get beyond the simplest level of control. INPUT will take you beyond... On the next few pages, there's a guide to some of the fascinating subjects that you'll be able to learn about in future issues.

- INPUT's structured guide ensures that your knowledge builds up issue by issue, so that you learn at your own pace. But you'll

INPUT lets you into the secrets of the professional programmers. And you won't just be learning dull facts and utility programs. There'll also be an insight into your computer's limitless potential for fun.

INPUT is full of ideas for graphics and games programs. There are lots of routines for popular graphic characters, plus the knowledge you need to create your own. And there's a guide to the techniques of games programming — movement, scoring, strategy



and adventure themes. Many of these are in the form of short routines that you can store on tape and add to your own ideas to create games and screen pictures of growing subtlety and skill.



BASIC PROGRAMMING

This is where INPUT gives you the nuts and bolts of writing your own programs. BASIC is a computer language that's easy to understand and easy to use. That's why all home computers are designed to use it, and that's why INPUT ensures that you'll learn how to get the most from all the commands that are at your disposal. Each issue includes interesting programs to try out what you've just learned, or to make better use of what you already know.

GAMES PROGRAMMING

Here's how you can write your own games programs and join in the fun with fresh ideas of your own. Each issue contains games routines that you can try at once, plus ideas that you can save up and add to your growing skills

MACHINE CODE

Learn to tackle your computer on its own level by understanding how it operates and how to get it to work to your advantage. No dull subject, this – machine code is the almost automatic choice for top class games programming. INPUT takes you through the techniques and the theory stage by stage so you don't need to digest huge slabs of boring mathematics. Starting straight away, there are plenty of routines to practise and put your skills to work

APPLICATIONS

What are home computers for? This is the section where you will find out, with examples of programs that you can type in right

away to start putting your computer to use. Whether you're interested in how computers can be used to store information, or want to try business programming or text editing, INPUT gives you the answers, plus lots more ideas.

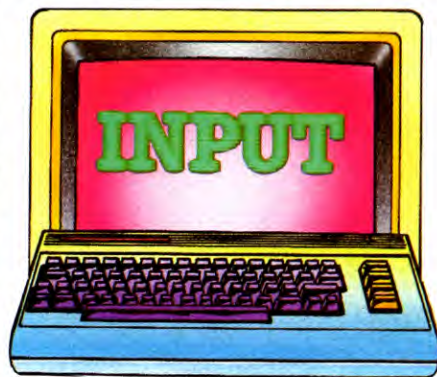
PERIPHERALS

An occasional series that shows you how to get the most out of computer add-ons. There's information on storage systems, joysticks for games control, printers

PLUS... HINTS AND TIPS

Whatever the subject, there's always bound to be more that you would like to know. So INPUT's packed with useful hints and tips, plus answers to many of the questions that so often just get overlooked. You'll find them in each of the sections above, clearly marked in colour so you know straight away where to look

On the next few pages, you'll find a guide to many of the fascinating subjects you'll see featured in each of INPUT's five departments – a course that will help you through every stage of programming.



also be able to put it to use, at once. It has been written to include eight of today's most popular home computers, so you can be sure that what you are learning is not just the theory – it's practical programming that will help you to get to know your own machine better. And at the same time you'll become familiar with computing in general – vital if you ever want to work with computers.

● Each issue of INPUT is divided into five departments that cover the different aspects of learning to use a computer.

HOW TO GET THE MOST OUT OF INPUT

There are some aspects of learning to program a computer which are common to all machines – like learning to understand binary, for example. But some things which will work on one computer won't work on another.

INPUT is written specifically to suit five popular home computers – the Sinclair ZX Spectrum, Commodore 64, Acorn Electron and BBC B, and the Dragon 32. Many programs are also suitable for the Sinclair ZX81, Commodore Vic 20 and Tandy Colour Computer in 32K.

This means that while there is general text explaining systems and procedures which are common to all the machines,

there are also sections in most articles which are specific to your own computer.

To help you to find your way around, each computer has been given its own symbol, and you'll find these wherever there is a section for a particular machine. On the back page of this guide there is a handy reference chart.

The program listings themselves have been specially designed and printed for clarity, as it is vital that they are typed in exactly as written. Even the smallest error, such as a comma that's missed out, can upset the whole program and prevent it from running properly. All the programs in INPUT have been checked on the computers for which they were written, so if you find that a program doesn't work first

time, the first thing you should do is to check that it has been entered into the computer exactly as printed.

You won't find many lengthy sections of text in INPUT. That's because learning to use a computer is best done by using the computer, not by reading about it in isolation. So a majority of the text is concerned with explaining a piece of programming that you have just tried out. That's why you'll find two typefaces in the text. BASIC keywords and commands – words like RUN, PRINT, GOTO – are printed in the same type whether they appear in the text or in a program listing, and this is different from the rest of the text. And when the text is telling you to press a particular key, you'll see it printed like this: RETURN.

BASIC PROGRAMMING



The best possible way to learn good programming is by writing good programs yourself! Here's where **Input** shows you the techniques at their most fundamental, by giving you the basics of **BASIC**, the easy to understand high-level language that all home computers use.

You'll learn all about the different commands, statements or functions that your computer recognizes – what each one does and when to use it. And since there's no point in learning the theory if you don't know how to apply it, each article includes lots of programs for you to try out, then modify...and loads of ideas for your own experiments.

Beginning with simple programs and working progressively up to longer ones, **Input** shows you how to put your knowledge together. Among the topics you'll encounter are:

IF... THEN

Two key words that make the computer make a decision

FOR... NEXT

A simple device to make the computer do part of a program over and over again

STEP

The program statement that controls the rate a loop is executed

GOTO

Make the computer skip to any line you specify

Subroutines

Small, self-contained parts of program designed to perform one function several times

Variables and strings

The way computers deal with numbers and words

Slicing strings

How words, sentences and other data can be cut up and put back together

INPUT

How to get the computer to ask you questions

PRINT

How to get the computer to give you answers

READ... DATA

How to feed information into your computer

RND

Get your computer to throw dice or pick a number

Concatenation

A valuable tool to help you write a letter

Number crunching

Taking the drudgery out of arithmetic, by getting your computer to do the sums for you

Dry run and TRACE

Squeeze the bugs out of your BASIC programs

Arrays

How to input large amounts of data in one go

NEW

The command that clears out one program and lets you start again

REM

How to 'flag' parts of the program so you remember what you're doing

ASCII codes

The standard codes that computers use to communicate with each other

CHR\$, VDU

How to call up the ASCII characters and print them on the screen

RUN

The command that starts the program running

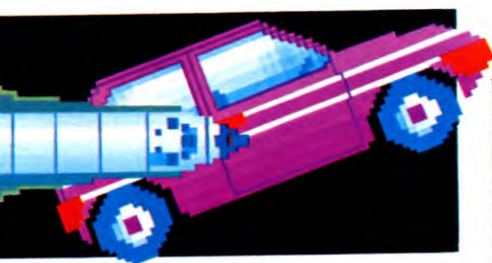
RESTORE

The statement which enables the computer to deal with more than one set of data at a time

GET, GET\$

Two ways to call up information at the touch of a key





INKEY, INKEYS

Two ways to feed in information at the press of a key

PEEK

Look directly into the computer's memory via its own assembler

POKE

Feed information directly into specific locations in the computer's memory

MODE

The statement used to select which display mode the computer is about to use

MOVE

How to move the graphics cursor to any point on the screen you specify

DRAW

The statement that defines the end of a line the computer is to draw

PLOT

The multi-purpose point, line and triangle drawing statement in BASIC

POINT

How to find out the colour of any position on the screen

RENUMBER

How to renumber your program lines

SAVE

The command given to SAVE a program

VERIFY

How to make certain that the program has been stored properly

DEF FN

Tell the computer that a function is about to be defined

GCOL

Set the colour that you want to use in your graphics

ELSE

What the computer should do if a condition has not been fulfilled

AND

How to extend the conditions when the computer is making decisions

TIME

Set or read the computer's internal timer

LOTS OF TIPS AND HINTS TO HELP MAKE IT EASY

Look for these fact-packed panels throughout the pages of Input. They're your handy guides to get you straight to the heart of computing problems

Microtip – the panel that shows you how to streamline your programs and make sure that they RUN. There are tips on the differences between individual computers, and the way in which they handle languages like BASIC and machine code.

Q & A – here's where you'll find short answers to baffling problems, plus explanations of new ideas, computer terms and systems. You'll find useful information on common errors, alternative methods you might like to try, and a whole host of facts to extend your knowledge.

Troubleshooter – for all those frustrating times when your programs just won't work. There are lots of ideas for how to avoid 'bugs' in your programs, and how to cure them. They're quick and they're simple, and they can be used time and again. Also in Troubleshooter – how to make sure that your peripherals are suitable and avoid 'crashing' through incompatibility problems.

your own program...
this series. We shall show you, too, how to create new graphics of your own.

Microtip

Watch the numbers!

The numbers which make up machine code routines must be typed into your computer with great accuracy. Once a routine is in memory, you can normally recall it for amendment.

Q+A

What's a sprite?

A *sprite* is a kind of high resolution user defined graphic (UDG), also referred to as a *movable object block*. It is used as a kind of high resolution graphic.

TROUBLESHOOTER

- Use a simple, reliable mono portable recorder and, if you can, reserve this for exclusive use with the computer. Always use mains power to ensure constant motor speeds
- Avoid using sophisticated stereo recorders unless its facilities can be switched out and mono playback is possible
- Use good quality audio or data tape and a brand which is reliable, and a brand which is well-worked-out

APPLICATIONS

Here's where to find out how to make your home computer earn its keep, with lots of programs to use at home or for your hobbies.

If you've got the aptitude, Applications points the way to a career in commercial programming, by showing you what's involved in business programs, datafiles and control systems. But if you just want to exploit your computer's potential as a powerful tool, then there's plenty here that you can use right away.

Learn how to store and retrieve your own information. Write a simple text editor or create an aid to planning your expenditure. You'll find all these, and lots more, in Applications:

Personal expenditure records
and bank account management

Recipe files
and menu planning

Letter writing
with simple text editors



Investment analysis
and keeping track of your money

Special datafiles
for your hobbies

Sight and sound programs
for pre-school teaching

The micro as business tool
What it can and can't do

Mailing lists
for business or for your club activities

Wordprocessing
for serious applications

Financial planning
using spreadsheets

Computer-aided learning
to supplement normal schooling

Touch typing teacher

Household inventory
for record keeping and insurance



Protecting your home
with watchdog programs'

Cashflow analysis

Music making
– composing tunes and simulating musical instruments

Club membership records

Cataloguing your collections

Creating an electronic diary

Household finance control

Statistical analysis

Modems

– linking your computer to the outside world via the telephone

MACHINE CODE



Although BASIC is easy to learn and a valuable tool for all kinds of jobs, it's not long before every would-be programmer learns its limitations for some applications. That's when you need to learn to use machine code, a powerful language that's anything up to 100 times faster.

You'll find the speed of machine code comes into its own particularly for games

programs, although there are routines you can use to improve almost anything you write, with better sound, clearer graphics and far quicker movement.

Starting with short routines to speed up your BASIC programs, you'll gradually come to know your computer on its own level, through binary, hexadecimal arithmetic and the structure of the machine's memory. Techniques that you will come to grips with include:

Assembly language
What it is and how to use it

Machine code
What it is; theory outline

Numbers in different bases
Binary, decimal and hexadecimal explained

Registers in the CPU
Understanding a hard-working component

Addresses and the stack
Machine code's RAM and ROM

Negative numbers
1's and 2's complement

Flags
What are they and how to use them

Logic
ANDs, ORs, EORs and XORs

Addition and Subtraction
in machine code

Shifting and rotating
Simple multiplication and division

Addressing a computer properly
Subroutines
GOSUBs AND RETURNs in machine code

Fast-moving graphics
for your BASIC programs

Sorting
Getting things in order with fast machine code sorts

GAMES PROGRAMMING

Fortunes have been made from top-selling computer games. But whatever level of programming skill you aim to reach, games are the part of programming that everyone really enjoys.

Here's where Input shows you how to program some simple games routines, and gives you the building blocks you need to make up your own completely original games. There's lots of ideas for high-resolution games graphics and exciting sound effects. You'll learn how to control movement around the screen, how to fire missiles, cause explosions, keep score and keep time.

And whether you're interested in space action games, strategy and tactical adventures, or want to adapt your favourite board game for the screen, Games Programming shows you the way. Subjects covered include:

Animation

How to simulate life and movement in your games characters

Keyboard control

Make your characters move left, right, up or down – or wrap around the screen

Timing

Use your computer's built-in timing mechanism to add interest to your games

Scoring

How to add scoring and 'new high' routines to your games

Shooting

How to fire missiles, laser beams or death rays at your target

Explosions

... and how to make your 'enemy' explode – or disappear – when you hit him

Sound effects

Making games more enjoyable with music and other types of sound



Graphics

Using user-defined graphics and ROM graphics in your games

Debugging games

Some of the things that might go wrong – and how to correct them

Planning a game

What you need to do to make a game good to play – and to sell

Random numbers

Use them to plot obstacles, move graphics, pick a card or make aliens unpredictable

Creating a character

Converting the characters from your own imagination into a screen display

Checking for obstacles

Has your character collided with anything? Is he going to be eaten by aliens?

Detecting a hit

Was the missile on target? And has it destroyed the enemy?

Title pages

Start your game impressively with big lettering and bold graphics

Card games

How to program your computer to play you at cards

Mazes

Creating maze games and mazes for arcade-type games

Word games

Quizzes on your computer

Dice games

Using the computer as a 'die shaker'

Theme for adventure games

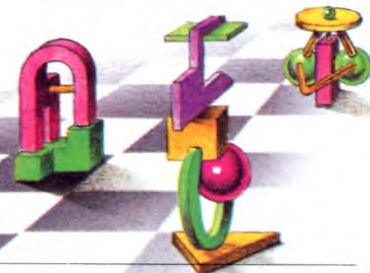
Dungeons and dragons? Spy-catching? A mystery? How to choose your theme

Mapping out an adventure

Using the keyboard to design your own fantasy world

Objects and obstacles

Creating perils and rewards in adventure games



PLUS PERIPHERALS



An occasional series in which we help you get trouble-free operation from computer add-ons – from the cheap tape recorder to the expensive disk drive or daisy-wheel printer.

INPUT-FOR YOU AND YOUR COMPUTER

LOOK FOR THESE SYMBOLS IN THE TEXT

**SINCLAIR ZX
SPECTRUM**



**SINCLAIR
ZX81**



**COMMODORE
64**



**COMMODORE
VIC 20**



**ACORN
ELECTRON
AND BBC B**



DRAGON 32



**TANDY
COLOUR
COMPUTER**



a much better way of teaching, as well as of programming.

Particularly when you are learning programming, it is always best to work out the 'core' of a program before adding the frills. So first try these lines (remember NEW!):

SS

```
10 LET N=INT (RND*12+1)
20 PRINT "WHAT IS □"; N; "□TIMES 9?"
30 INPUT A
40 IF A=N*9 THEN PRINT "CORRECT"
```

CE

```
10 LET N=RND(12)
20 PRINT "WHAT IS □; N; □TIMES 9?"
30 INPUT A
40 IF A=N*9 THEN PRINT "CORRECT"
```

CE

```
10 N=INT (RND(8)*12)+1
20 PRINT "WHAT IS □"; N; "□TIMES 9?"
30 INPUT A
40 IF A=N*9 THEN PRINT "CORRECT"
```

This program is using RND in much the same way as in the guessing game. In Line 10, you first set up a variable, or label, for the random number the computer selects. In this case it is N, but it could be any other letter or letters.



3. Computer symbols for basic arithmetic. The * sign, and not small x, is used for 'multiply.' The / means 'divided by,' and ^ 'raised to the power of.' Plus and minus signs are conventional



by 9 whichever number chosen this time. Line 10, you first set up a variable, or label, for the random number the computer selects. In this case it is N, but it could be any other letter or letters.

You can start the course right away, as no previous programming knowledge is assumed. But you do need to know your way around the keyboard of your own machine — how to enter, edit and delete program lines, how to find the various *modes*, and so on. If you do not know this, and also how to **SAVE** and **LOAD** programs on tape, please read these sections of your computer's manual or introductory booklet before you start.

COPYING PROGRAM LISTINGS

It's absolutely vital that program listings are copied exactly when you type them into your computer — otherwise the program won't work properly, and sometimes it won't work at all. Because of this, all the program listings in **INPUT** are designed to be easy to read and to copy. If you have a problem with any of the programs, double-check that you have entered it as printed. Some of the main points to watch out for are as follows:

PUNCTUATION

Because punctuation is so important in computing, we've 'invented' a new typeface — clear, computer-like **SANS SERIF** lettering, but with the much bolder punctuation **“;,:”** of a conventional typeface. This is used throughout the program.

```
40 PRINT "HELLO, □":AS, "I HAVE SOME","
QUESTIONS FOR YOU"
```

SPACES

Spaces are indicated by empty boxes □ □ □ so you can see how many spaces are needed — important, sometimes, in graphics programs.

```
80 PRINT "WHAT IS □":N; "□TIMES 9?"
```

SYMBOLS

Each machine's standard symbols — for example = × — are printed as they appear on the keyboard.

```
10 PRINT "□"
```

DIAGRAMS

Diagrams for user-defined graphics, sprites and so on are divided by coloured rules so you can see clearly "what goes where" when you are typing in each line.

If you own a Sinclair ZX Spectrum (16K or 48K) or ZX81, Commodore 64 or Vic 20, Acorn Electron or BBC B, Dragon 32 or Tandy Colour Computer, then **INPUT** is for you. That's because it gives you an exciting new way to learn computer programming, with many programs written specially for your computer.

Inside this course synopsis, you'll find a guide to what you can look forward to in future issues of **INPUT**. But because **INPUT** is about learning to use your computer, not spending hours reading about it, here's a ready reference guide to putting the course to work for you.

To start with, you need to know where to find the programs and text that are specially written for your computer. **INPUT** makes it easy for you — each computer is identified by one of these clear emblems that you'll find whenever a section is specifically for your machine.

SHORT EMBLEMS



Emblems like this mean that the following few lines of program are for you. (You should also read the text in between!)

Grouped short emblems, such as



mean the program lines and text which follow are for all the computers whose symbols appear.

If all four symbols are used, they are for everyone.



LONG EMBLEMS



Long emblems mean that the whole of the following section, both text and program lines, is for your machine.