

# Chapter 5: Software

1

**What is software?**

**What is hardware?**

## Operating Systems:

(this controls the hardware)

(E.G. MS-DOS - Microsoft Disk Operating System)

The operating system controls the printer and other input and output devices, it communicates with other back store hardware (e.g. storing / reading data).

## User Interfaces

*This is what you see when the computer is switched on. It's like a welcome to the user. It has icons, menu's etc. It should be easy to use and obvious to a new user.*

*E.g At Ease*

**They are often command driven, menu driven or graphically driven.**

## Chapter 5: Software

2

### But what do they mean???

**Command Driven:** When you type in an instruction.

**Menu Driven:** Lists of options available.

**Graphically**

**Driven:** Icons. Apples were among the first to use Icons.

### How would you design a user interface?

It should:

- Be Consistent (all parts of the system should behave in the same way.)
- Be Consistent (the layout)
- Colour (should be easy to read)
- Sound (not annoying and must have the option to turn it off)
- Help (simple, clear and quick to use)

## Utility Programmes:

These programmes usually come preinstalled on the hard drive but it is possible to purchase them. They list files, delete files, copy files, send files to print, sort data and repair damaged files. E.G. Virus Checker

## Applications

What is an application????

Lots of different types are available:-

Word processing  
Spreadsheets  
Databases

## Chapter 5: Software

4

### Drawing

These packages are available separately or as *Integrated Software* (these programmes consist of all of the above all in one).

#### ADVANTAGE:

cheaper

easier to learn

#### DISADVANTAGE:

They are often not the best quality.

Some companies exist that write tailor made programmes. E.G. Baan

Most packages can be used by a number of different organisations E.G. Word processing can be used by Airlines and

## Chapter 5: Software

5

Zoo's.

### Computer Languages:

Computers can only understand

**BINARY CODE**

(01111001)

all languages eventually have to be reduced to this.

**Low-level Languages: Simple for the computer but difficult for us.**

**Machine Language: 1's and 0's. Often machine specific. Does not transfer between computers.**

**Assembly Language: Uses simple language, easier for the programmer.**

**High- Level Language:** Not machine dependent, so once a programme has been written it can be transferred. It is written in a language similar to English and uses instructions such as *PRINT*, *GOTO* and *READ*.

**Advantages:-** Simple Instructions  
Easy to correct errors  
Transferable programmes

**Translation Programmes:**

These are used to convert programmes into machine codes.

**Compilers And Interpreters:**

These programmes change high-level language into machine code.

## Chapter 5: Software

**An interpreter:** this is like looking at each individual word rather than a whole document. The Source will be translated line-by-line. Example: Basic

**Compiler:** Takes the whole programme (source program) and translates it in one go. A compiler needs more time to translate, but will make a more efficient machine program (and will produce better error files)

**Assemblers:** often one instruction in assembly equals one in machine code instruction. Assemblers are low-level language translators.