

## Programming techniques

- **Sequence**
- **Selection**
  - IF... ELSE...
- **Iteration**
  - For & While
- **Basic string manipulation**

### file handling operations:

- open
- read
- write
- close
- **the use of records to store data**
- **the use of SQL to search for data**

### Arrays

- one dimensional arrays
- two dimensional arrays

### Sub programs

- Functions
- Procedures

### Data types

- **Integer** e.g. 23
- **Real** e.g. 23.7
- **Character** e.g. A or 5
- **String** e.g. A546TH
- **Boolean** e.g. TRUE or FALSE.

## Arithmetic operators

+	Addition e.g. $x=6+5$ gives 11
-	Subtraction e.g. $x=6-5$ gives 1
*	Multiplication e.g. $x=12*2$ gives 24
/	Division e.g. $x=12/2$ gives 6
<b>MOD</b>	Modulus e.g. $12\text{MOD}5$ gives 2
<b>DIV</b>	Quotient e.g. $17\text{DIV}5$ gives 3
^	Exponent e.g. $5^3$ gives 125

## Comparison operators

= or ==	Is equal to
>	Is greater than
<	Is less than
<> or !=	Is not equal to
>=	Greater than or equal to
<=	Less than or equal to

## Robust programs

- defensive design considerations:
  - input sanitisation/validation
  - planning for contingencies
  - anticipating misuse
  - authentication
- maintainability:
  - Comments & Indentation

- Purpose of testing
- Types of testing
  - Iterative
  - Final / terminal
- Selecting and using appropriate test data
- Logic errors
  - Definition & examples.
- Syntax errors
  - Definition & examples.

## Translators & facilities of languages

### Low level languages:

- Machine language
  - Op-code
  - Operand
- Assembly language
  - Mnemonics

### High level languages:

- Source code
- Assembler
- Compiler
- Interpreter

### Integrated development environment (IDE).

- Source code editor.
- Error debugger.
- Run time environment.
- Translator (compiler or interpreter).
- Automation tools