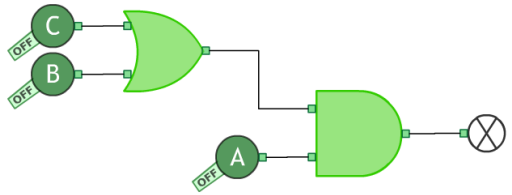
Year 10 Computer Science KO
2.4 Computational logic

An example logic circuit, with notations and truth table

$$A \wedge (B \vee C)$$



A	B	C	B V C	A ∧ (B V C)
0	0	0	0	**0**
0	0	1	1	**0**
0	1	0	1	**0**
0	1	1	1	**0**
1	0	0	0	**0**
1	0	1	1	**1**
1	1	0	1	**1**
1	1	1	1	**1**

Careers

- Software development
- Programing
- Software Engineering

Why is data represented in Binary in a computer system?:

- Binary is the representation of the 'presence' of electricity
- If present or 'on' we use a: 1
- If absent or 'off' we use a: 0
- We can use this idea to change 1 and 0 states through the use of logic gates
- Logic Gates take inputs and covert them to an output
- The digit 1 or 0 is stored in transistors which are in the processor.

Half Adder: A circuit to add two binary bits together

Full Adder circuit: 2 half adders together. This can add more than two binary bits together and deal with carrying bits over

Year 10 Computer Science KO
Programming Techniques

Careers
- Software development
- Programing
- Software Engineering

Constant
Value STORED IN A **MEMORY LOCATION** that **never changes WITHIN A PROGRAM**

Variable
Value STORED IN **MEMORY LOCATION** that **can change WITHIN IN A PROGRAM**

Syntax Error
An error in the rules/grammar of the language Eg missing colon / spelling mistake

Logic Error
The program is written to do something other than what the programmer intended Eg Resetting only the first 9 elements in an array instead of all 10.

Run Time Error:
More difficult to spot as it can run a program without reporting an error. E.g. runs but doesn't give an output. Or the program hangs or Becomes inactive

Sequence: Completing steps in the order which they must happen

Selection: Where a choice is made in a program depending on a condition or outcome

Iteration: Act of repeating or lopping specific sections of code

Count controlled Iteration: Repeats a set number of times

Condition controlled: Repeats until a condition is met or something in the program changes

Data Types

Real /Float
Number with decimal Point

Integer
Number without a decimal Point

String
A series of characters/TEXT

Character
A single letter or symbol

Date/Time
Date and Time in any format

Boolean
Yes no, true false value

Other Info

Concatenate
To join different data types together

 # Comments
Use these to add comments in to your code to explain what you have done

Validation: An computer check to ensure that the data entered is sensible and reasonable. It does not check the accuracy of data.

Year 10 Computer Science KO – Programming Techniques

Comparison Operators

==	Equal to
!=	Not equal to
<	Less than
<=	Less than or equal to
>	Greater than
>=	Greater than or equal to

Aritmetic Operators

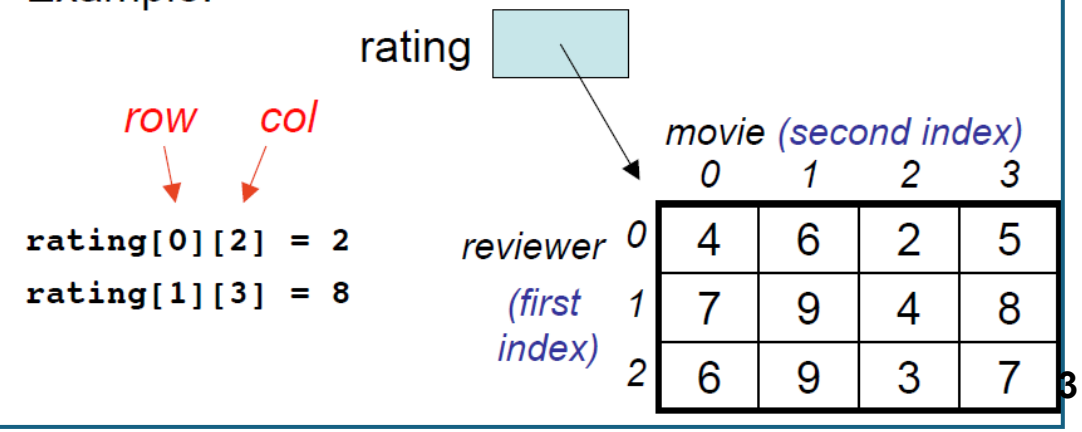
+	Addition eg x=6+5 gives 11
-	Subtraction eg x=6-5 gives 1
*	Multiplication eg x=12*2 gives 24
/	Division eg x=12/2 gives 6
MOD	Modulus eg 12MOD5 gives 2
DIV	Quotient eg 17DIV5 gives 3
^	Exponentiation eg 3^4 gives 81

Careers

- Software development
- Programing
- Software Engineering

TYPE	INFO	SYNTAX
LIST	MUTABLE DIFFERENT DATA TYPES	[ ] E.G. [1,"HELLO", 3.4]
TUPLE	IMMUTABLE DIFFERENT DATA TYPES	( ) E.G. (1,2, "Hello", 4.3)
ARRAY	IMMUTABLE SAME DATA TYPE	[ ] E.G [1,2,3,4]

- Two-dimensional (2D) arrays are indexed by two subscripts, one for the row and one for the column.
- Example:



Year 10 Computer Science KO – 2.1 Algorithms – Computational Thinking

What is Computational thinking?

The thought processes involved in formulating a problem and its solution(s), so that a computer, human or machine can effectively carry out

1

How do you think computationally?

To effectively solve problems you need to….

- Decompose
- Abstract
- Algorithmic thinking
- Create algorithms

1

KEY TERMS

Algorithm: Steps to provide a solution to a problem, usually represented in flowcharts or pseudocode

Decompose: **B**reaking down a large problem into smaller sub-problems

Abstraction: Representing 'real world' problems in a computer using variables and symbols and removing unnecessary elements from the problem

Algorithmic Thinking: Identifying the steps involved in solving a problem.

Sequence: Completing steps in the order which they must happen

Selection: Where a choice is made in a program depending on a condition or outcome

Iteration: Act of repeating or lopping specific sections of code

2

Careers

- Software development
- Programing

3

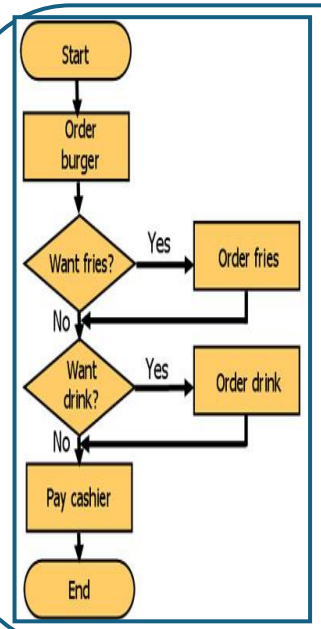
Flowcharts

Displays an algorithm in diagram form using symbols and arrows to show to flow of information

3

Pseudocode

A structured use of English used to define the steps needed to solve a problem.




```
WHILE TRUE:
temp = “”
IF temp > than 20c THEN
        Open windows AND heaters OFF
ELSE
        Close windows AND heaters ON

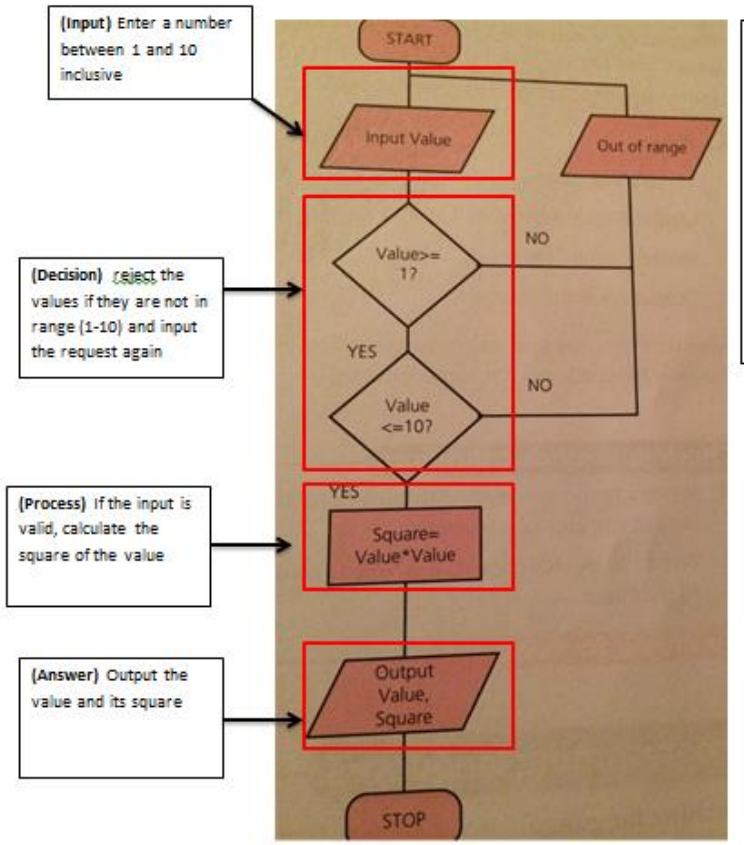
TIME = “”
IF TIME = 18.00  THEN
        Sprinklers on
ELSE
        Sprinklers off
        Continue
BREAK
```

3

4

Year 10 Computer Science KO – 2.1 – Algorithms - Flowcharts

Careers
- Software development
- Programing
- Software Engineering



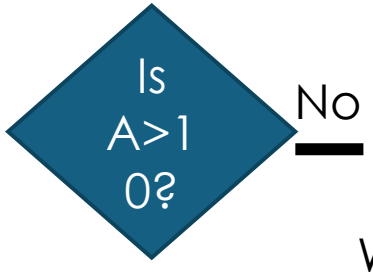
START/STOP

Always start and end with this

INPUT/OUTPUT

Use when there is an input or output required e.g. user inputs their name, program displays their name

Decision



When a **choice** has to be made in the program

SUB ROUTINE

Sequence that performs a specific task. You can use this within your flowchart to show more detail in a specific section

PROCESS

To do something in the program e.g a calculation



Flow lines – show the flow of information in the algorithm

Year 10 Computer Science KO – 2.1 Algorithms – 5. Pseudocode

Pseudocode uses English. It mimics how your code may look in the programming language BUT it DOESN'T have to be exact

Some useful terms you could use.

START
IF
ELIF
ELSE
FOR
WHILE TRUE
ENDIF
END

Here are some symbols to use

Comparison operators

== Equal to
!= Not equal to
< Less than
<= Less than or equal to
➢ Greater than
>= Greater than or equal to

Arithmetic Operators

+ Add
- Subtract
/ Divide
*** Multiply**
MOD will return the remainder for the division

Careers

- Software development
- Programing

```
WHILE TRUE:
temp = ""
IF temp > than 20c THEN
        Open windows AND heaters OFF
    ELSE
            Close windows AND heaters
    ON

    TIME = ""
    IF TIME = 18.00  THEN
            Sprinklers on
    ELSE
            Sprinklers off
            Continue
    BREAK
```

This repeats, it reads the temp of a greenhouse. If greater than 20 degs then open windows and turn heaters off ELSE close windows and heaters on

It checks the time if the time is 6pm then turn sprinklers on otherwise keep sprinklers off

Year 10 Computer Science KO – 2.1 – Sorting Algorithms – Bubble, Insertion & merge

Key vocabulary	
Bubble sort	Works by repeatedly going through the list to be sorted, comparing each pair of adjacent elements. If the elements are in the wrong order they are swapped, else they are left in position.
Insertion sort	Sorts data one element at a time. The algorithm takes one data item from the list and places it in the correct location in the list. This process is repeated until there are no more unsorted items in the list. More efficient than bubble sort.
Merge sort	This is a two-stage sort. Firstly the list is split in half into sub lists repeatedly. The algorithm stops splitting the lists when each list has only 1 element in it. The second stage involves repeatedly merging the lists in order until there is only one sub list remaining.

5 1 12 -5 16	unsorted
5 1 12 -5 16	5 > 1, swap
1 5 12 -5 16	5 < 12, ok
1 5 12 -5 16	12 > -5, swap
1 5 -5 12 16	12 < 16, ok
1 5 -5 12 16	1 < 5, ok
1 5 -5 12 16	5 > -5, swap
1 -5 5 12 16	5 < 12, ok
1 -5 5 12 16	1 > -5, swap
-5 1 5 12 16	1 < 5, ok
-5 1 5 12 16	-5 < 1, ok
-5 1 5 12 16	sorted

Figure 1 - Bubble sort example

7 -5 2 16 4	unsorted
7 -5 2 16 4	-5 to be inserted
? 7 2 16 4	7 > -5, shift
-5 7 2 16 4	reached left boundary, insert -5
-5 7 2 16 4	2 to be inserted
-5 ? 7 16 4	7 > 2, shift
-5 2 7 16 4	-5 < 2, insert 2
-5 2 7 16 4	16 to be inserted
-5 2 7 16 4	7 < 16, insert 16
-5 2 7 16 4	4 to be inserted
-5 2 7 ? 16	16 > 4, shift
-5 2 ? 7 16	7 > 4, shift
-5 2 4 7 16	2 < 4, insert 4
-5 2 4 7 16	sorted

Figure 2 - Insertion sort example

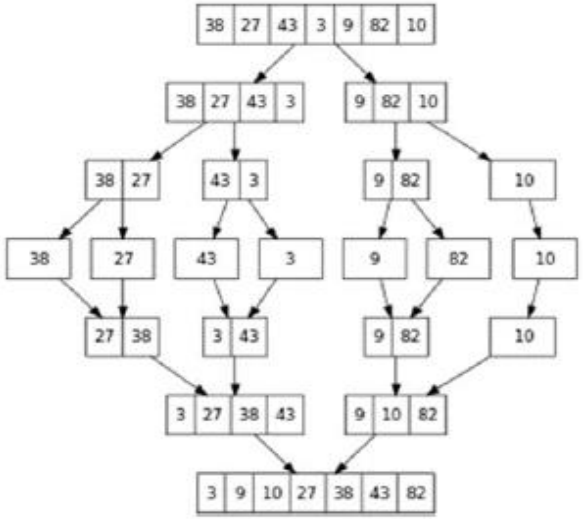


Figure 3 - Merge sort example

Careers

- Software development
- Programing
- Software Engineering

Year 10 Computer Science KO –
2.1 Sorting Algorithms – Binary and Linear Searches

Key vocabulary	
Linear Search	Data may be in any order to complete a linear search. Each item is inspected in turn to see whether it is what is being searched for. If an item is found, then True is returned, else the next element is inspected until all items have been searched. If nothing is found by the end of the algorithm then False is returned.
Binary Search	If a list is sorted (numerical or alphabetical order) then a more efficient algorithm can be used. It works by repeatedly dividing the list into half and searching in the appropriate half.



Figure 1 - Linear search example

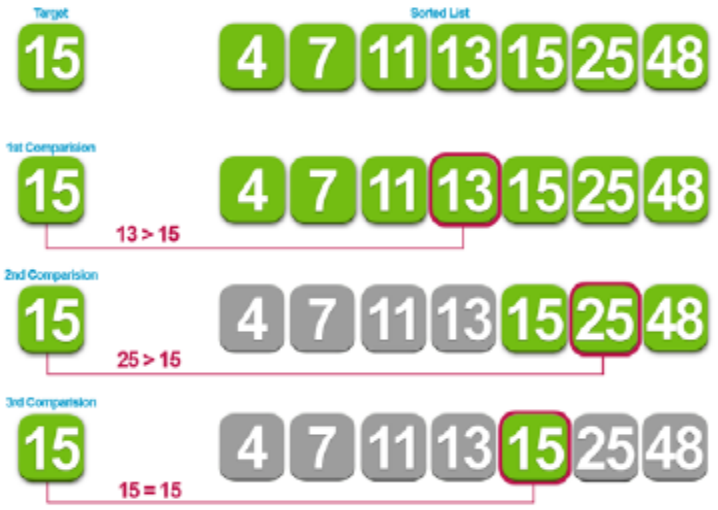


Figure 2 - Binary search example

Careers

- Software development
- Programing
- Software Engineering

Year 10 Computer Science KO - 2.5 Translators and Facilities

High Level Language (eg Python)

- Use Instructions are in Words
- Designed to be read by human programmers
- Portable/translated for different machines

PROS: Easy for humans to understand as written in English

CONS: Needs to be translated in to machine code so not as fast to execute

Machine Code (Binary Code) (low level)

- Instructions are in binary code Designed to be read/executed by the computer
- Specific to a particular machine

PROS: Fast no need to translate already in binary, close to CPU

CONS: Difficult to read and understand for humans

Assembly Language (low level)

- Low Level Language written in mnemonics that closely reflect the operations of the CPU
- Eg LOAD

PROS: A bite easier to understand than binary as uses mnemonics

CONS: Still needs translating to machine code using an Assembler translator

How Instructions are stored in Binary.

- The instruction consists of an operator/op code and an operand
- both stored as bit patterns
- (op code) from a given instruction set
- Each op code has a unique bit pattern

Why Use Binary.

- So that computers can be based on logic circuits.
- (each part of the circuit) can be in one of two states
- 0 and 1/true or false

Translators

Interpreters:

Translates **one line of HL** code at a time…

- … and executes it
- … stops when it finds an error
- … can be resumed

Compilers:

Translates **whole program of HL code** at a time…

- … and executes it
- … stops when it finds an error

Assembler:

Translates **Assembly language** into machine code

Selection

A condition is used to decide executed

- if x = 1 then y = 3 else x

whether code should be = 2

Sequence

A list of instructions to be followed one after the other. Step by step

Iteration

code is executed repeatedly

```
For I in range (10);
    Print (i)
```

Careers

- Software development
- Programing
- Software Engineering

Year 10 Computer Science KO - 2.5 Translators and Facilities

TOOLS FOR PROGRAMMING

IDE (intergrated Development Environment): A software application that provides all the facilities to computer programmers for developing programs. It normally consists of a source code editor, build automation tools and a debugger

Code Editor – edits program text, you can type our your source code here

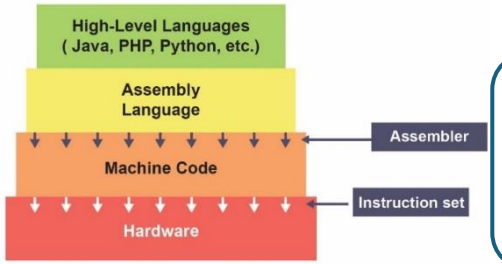
Syntax Checks – Highlights syntax errors

Runtime Environment – allows programs to be run one line at a time – helps test programs and locate errors.

Translator – Compiles or interprets the code

Libraries – Provides functions not included in core part of programming languages (eg Random)

Debugger – Helps to detect errors. Suggests what type of error it is and what line it is on.



Careers

- Software development
- Programing
- Software Engineering

Source Code: The code the programmer writes

Object Code: code converted by the compiler so that it can be understood by the computer

Executable Code: Files which contain **instructions in machine code.** These instructions are **carried out in the computer hardware**

Programming Standards
Code should follow agreed conventions (EG Lowercase for variable names, schemes to be followed).
Language code is written in.
Functions used to tidy up repeated code.
Comments explain the code clearly.
Correct use of indentation.
Useful identifiers (File names & Variable names)
Code should follow agreed conventions