

Year 10 Computer Science – Spring Term

Unit 2 – Writing algorithms (Pseudocode)

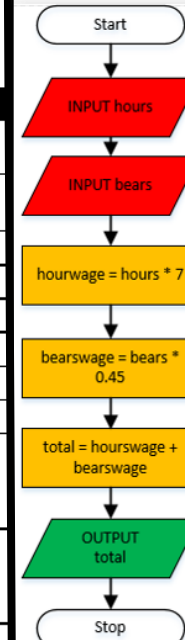
Pseudocode

This means 'fake code'. It's part way between English sentences, and programming code. It is language neutral (it can be read by programmers who are able to use any language).

Examples of pseudocode syntax and explanations

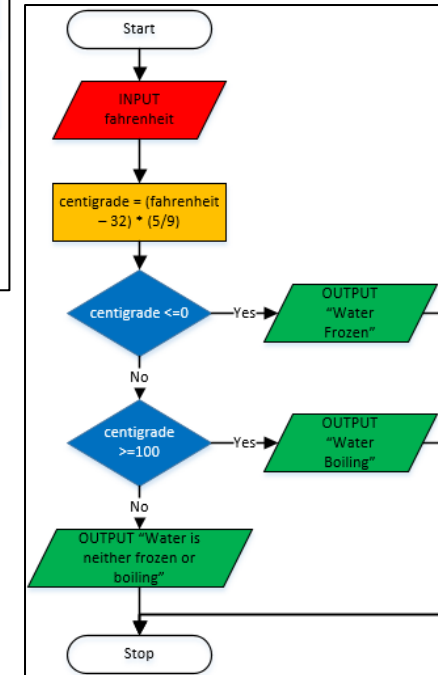
<code>x = 5</code>	Declares a new variable called x and gives it a numerical value of 5
<code>name = "Bob"</code>	Creates a new variable called name and sets its value to "Bob"
<code>str(x)</code>	Casts the value in x to be a string value
<code>int(x)</code>	Casts the value in x to be an integer value
<code>float(x)</code>	Casts the value in x to be a float value
<code>print(name)</code>	Prints a variable to the screen
<code>print("Hello")</code>	Prints the given string in quotes to the screen.
<code>name = input("Please enter your name")</code>	An input from the user which asks them for their name and stores in a variable.
<code>for i = 0 to 7</code> <code>print("Hello")</code> <code>next i</code>	A count controlled loop which will print "Hello" 8 times (0-7 inclusive).
<code>while answer != "Computer"</code> <code>answer = input("What is the password?")</code> <code>endwhile</code>	A condition controlled loop which asks a user for a password until they correctly guess with "Computer".
<code>do</code> <code>answer = input("What is the password?")</code> <code>until answer == "Computer"</code>	A condition controlled loop which asks a user for a password until they correctly guess with "Computer".
<code>entry = input("Enter a selection")</code> <code>if entry == "a" then</code> <code>print("You selected a")</code> <code>elseif entry == "b" then</code> <code>print("You selected b")</code> <code>else</code> <code>print("Unrecognised selection")</code> <code>endif</code>	Selection can be carried out to identify certain situations within a program. The program here takes an input and prints different statements for the A and B selection.
<code>function triple(number)</code> <code>return number * 3</code> <code>endfunction</code>	Creates a simple function to triple a number given as an input.
<code>array names[3]</code> <code>names[0] = "Ahmad"</code> <code>names[1] = "Ben"</code> <code>names[2] = "Catherine"</code>	Creates an array called names, the length is set to 3. Names are then added to the positions in the array.

Sequence



Programming Constructs

Selection



Year 10 Computer Science – Autumn Term

Unit 1 – Computer Systems, Memory and Storage

CPU- Fetches, decodes and executes instructions. Contains-

- **Control Unit- Decodes** instructions/ Sends **control signals** to control how data moves around the CPU.
- **Arithmetic Logic Unit-** does the calculations/ performs AND, OR, NOT and Binary Shifts/ contains the accumulator.
- **Cache-** fast access to **frequently used instructions** and **data** without having to go to the main memory (**RAM**).

CPU and system performance

Clock Speed

- Number of **instructions** a processor core can carry out per sec (Hz).
- Average 3.5GHz (3.5 billion instructions per second).
- **Higher clock**= more instructions.
- **Overclocked**= higher clock speed causes overheating and damage to system.

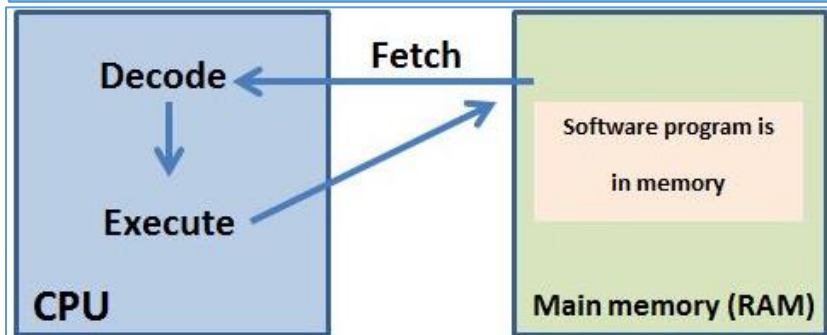
Number of cores

- Each core processes data independently.
- More cores means CPU can carry more instructions so faster processing.

Cache Size- Larger cache give CPU faster access to more data to process.

Von Neumann's Design Revolutionised Computing- describes a system where the CPU runs programs stored in memory. Programs consist of instructions and data which are stored in memory addresses.

- **Memory address register (MAR)** holds the memory address used by the CPU.
- **Memory data register (MDR)** holds **data fetched from or to be written to memory**
- **Program counter (PC)** holds the **address of the next instruction**. Once an instruction is fetched its value is increased by one.
- **Accumulator** stores intermediate results of calculations in the ALU.



Embedded systems are computers inside larger systems

- Examples include dishwashers, microwaves and TVs (**dedicated** systems).
- Used as **control systems** to **monitor and control** e.g. dishwasher controls water pumps and thermostat. ****Advantages- faster, cheaper, less power****

Memory

RAM (random access memory)-

- Main memory where data, files and programs are stored when used.
- When computer boots, operating system copied to RAM.
- Slower than Cache memory.
- Can be read and written to.
- It is volatile/ temporary memory, needs requires power to retain data.
- Portable entertainment games system needs some RAM to store progress data as the game is being played.

ROM (read only memory)-

- Non-volatile, can only be read NOT written to.
- Is on a chip built into the motherboard.
- Contains instructions to boot up- known as BIOS (basic input output system).
- A cartridge for a portable entertainment games system could be ROM because the instructions do not need to be changed.

Virtual Memory-

- When RAM full, application data that has not been used recently is put on virtual memory.
- Used when too many applications are open.
- If CPU needs to read data it moves it back to RAM.

Types of Secondary Storage

Flash - Fast speed of data transfer, no moving parts, requires little power, good for portable and modern devices



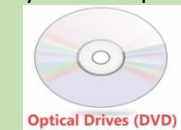
Flash Storage Drives (SSD)

Magnetic - Uses magnets to store data. Longer read/write life, large capacity for cheaper price, has moving parts so not good for portability.

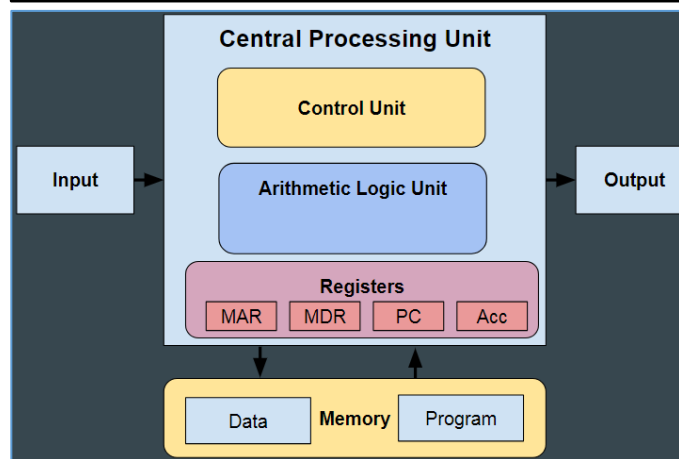


Magnetic Spinning Drives (HDD)

Optical - Uses a laser to make pits and lands in the surface. Scratched easily, but good portability and cheap.



Optical Drives (DVD)



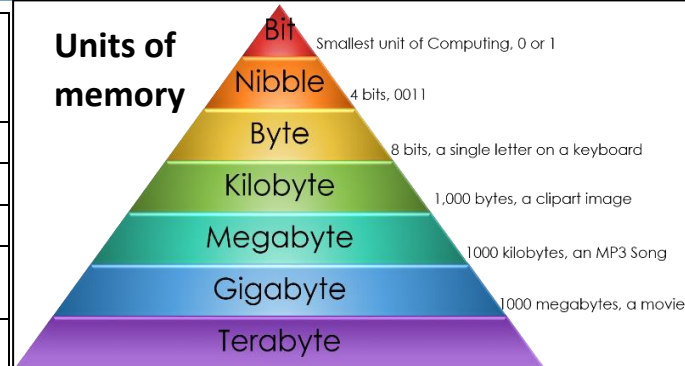
Characteristics to consider when choosing secondary storage-

- Capacity-** Maximum amount of data that can be held.
- Speed-** How quickly data can be accessed e.g. 10 gigabytes per second.
- Portability-** How easy it is to move from one computer to another.
- Durability-** how robust is the medium?
- Reliability-** how resilient and long lasting is the medium?
- Costs-** price per gigabyte or megabyte.

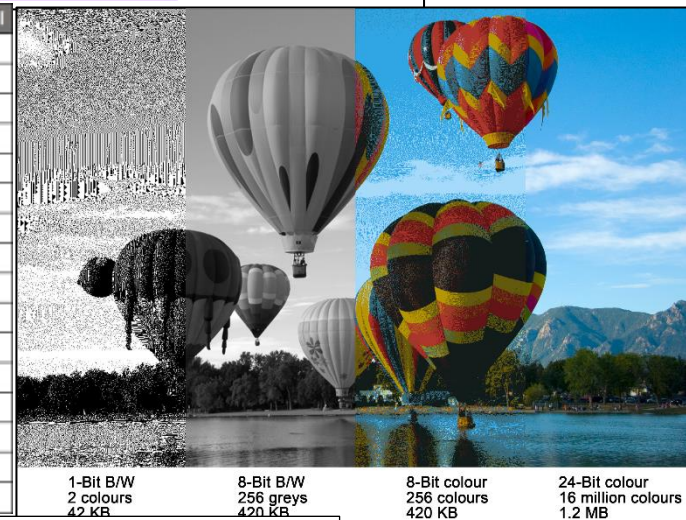
Year 10 Computer Science – Autumn Term

Data Representation

Character	Any single letter, number, space, punctuation mark, or symbol that can be typed on a computer.
Bit	Binary digit – 1 or 0
Nibble	4 binary digits e.g 0110 or 0001
Byte	8 binary digits, e.g. 01101100
Binary	Language used by computers to store and process data – base 2
Denary/Decimal	Number system used by humans 0-9 – base 10
Hexadecimal	Shortened version of binary. Number system using 16 characters – 0-9 and A-F – base 16
Character Set	Defined list of characters recognised by the computer
ASCII	Character set using 7 bits – covers the most common used English letters, numbers and symbols. Extended ASCII uses 8 bits.
Unicode	Character set using up to 32 bits – represents all possible characters across every language.
Bitmap	An image made up from a series of coloured dots (pixels)
Vector	An image made up of lines and shapes
Resolution	The number of pixels within a fixed area.
Colour depth	The number of bits used for each pixel – more bits = more colours
Sample rate/frequency	The number of samples taken per second.
Sample size	The amount of bits that are available for each sample.

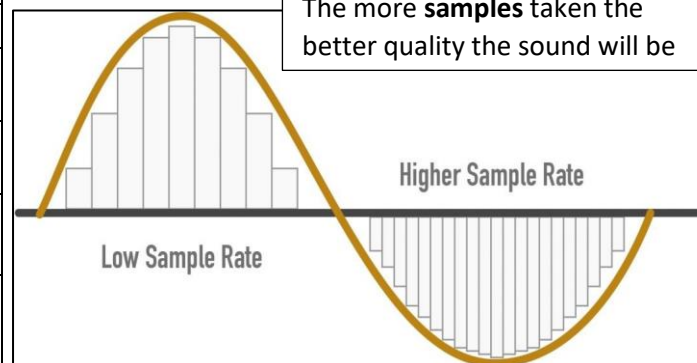
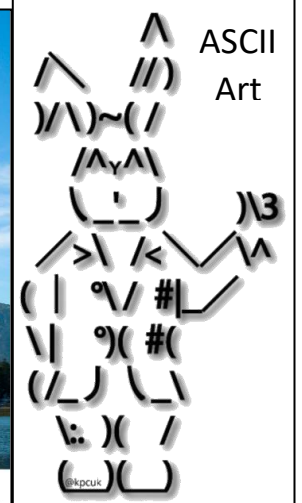


Decimal	Binary	Hexadecimal
0	0000	0
1	0001	1
2	0010	2
3	0011	3
4	0100	4
5	0101	5
6	0110	6
7	0111	7
8	1000	8
9	1001	9
10	1010	A
11	1011	B
12	1100	C
13	1101	D
14	1110	E
15	1111	F



Converting binary to denary

$$\begin{array}{ccccccc}
 128 & 64 & 32 & 16 & 8 & 4 & 2 & 1 \\
 \downarrow & \downarrow & \downarrow & \downarrow & \downarrow & \downarrow & \downarrow & \downarrow \\
 1 & 0 & 0 & 1 & 1 & 0 & 1 & 1 \\
 \hline
 128 + 0 + 0 + 16 + 8 + 0 + 2 + 1 \\
 = 155
 \end{array}$$



Helpful websites:

Youtube – Binary conversion

BBC Bitesize

<https://www.computerscience.gcse.guru>

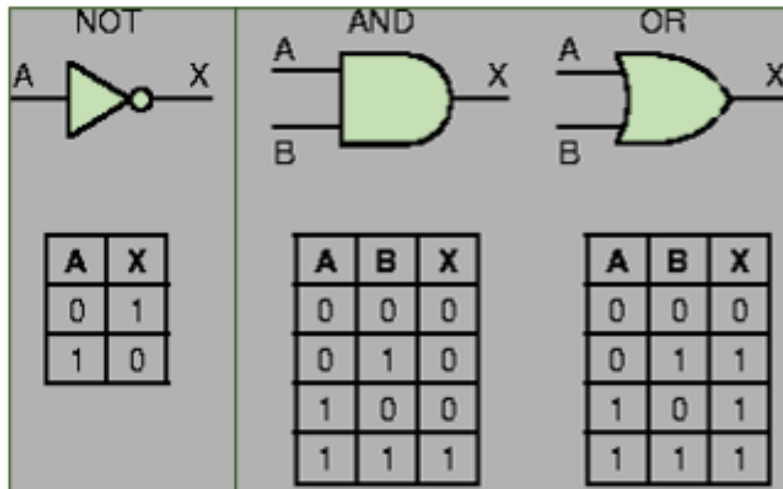
<https://games.penjee.com/binary-bonanza/>

Year 10 Computer Science – Spring Term

Computational logic

Logic gates

In its most basic form, a computer is a collection of powered and unpowered circuits and transistors. A logic gate is a series of transistors connected together to give one or more outputs, each output being based on the input or combination of inputs supplied to it.

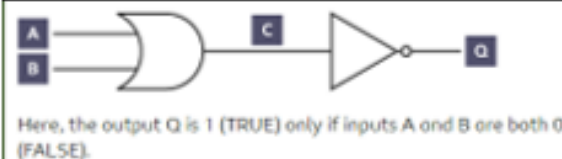


NOT – uses one input to generate one output. A NOT gate inverts the input - the output is opposite to the input.

AND – uses two inputs to generate one output. The output is 1 (TRUE) only if both of the inputs are 1 (TRUE).

OR – uses two inputs to generate one output. The output is 1 (TRUE) only if either or both of the inputs are 1 (TRUE).

Combining gates together



A	B	C	Q
0	0	0	1
0	1	1	0
1	0	1	0
1	1	1	0

Why data is represented in binary form

A computer is basically a collection of transistors and circuits. These components have 2 states:

- on - a current is flowing through the component
- off - a current is not flowing through the component

These two states can easily be represented by using binary:

- 1 = on (TRUE)
- 0 = off (FALSE)

truth table

A table to list the output for all possible input combinations into a logic gate.

Operator	Mathematical function	Example
+	Addition	1+2=3
-	Subtraction	4-1=3
^	Exponential	2^4=16
*	Multiplication	4*3=12, (5*6)=30
/	Division	12/4=3
Mod	Modulus (return the remainder from an integer division)	15 Mod 4=3 255 mod 10=5
\	Integer Division (discards the decimal places)	19\4=4

THE LOGIC LAB



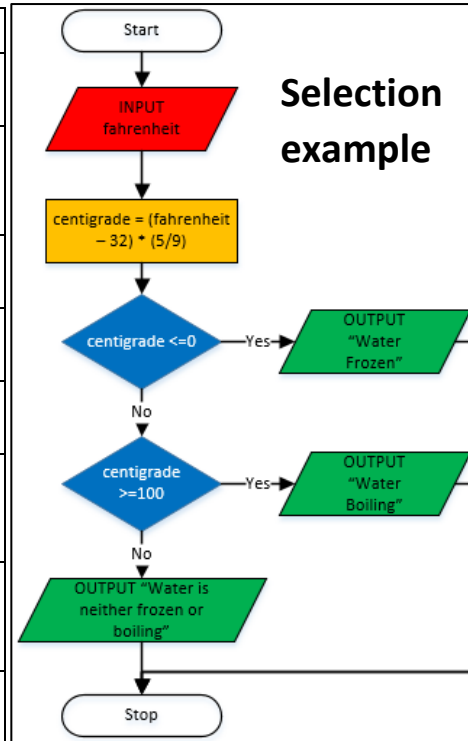
<http://www.neuroproductions.be/logic-lab/>

Can be used to create the gates – especially when doing combination gates and simulate using all the combinations of inputs and demonstrate what the output would be. This would also help you to complete the truth tables.

Year 9 Computer Science – Spring Term

Python Programming

Python	A high level programming language.
Programming	The process of writing computer programs. Code The instructions that a program uses.
Sequence	Parts of the code that run in order and the pathway of the program reads and runs very line in order.
Selection	Selects a pathways through the code based on whether a condition is true
Iteration	Code is repeated (looped), either while something is true or for a number of times
Algorithm	A set of rules/instructions to be followed by a computer system
Variable	A value that is stored in the program and can change whilst the program is executed. (eg. time, speed)
Arithmetic Operator	Used when calculating data. + - / * % (remainder of the calculation) ^ (to the power of)
Comparative Operator	When comparing data, an operator is used to solve the equality such as <>, != or ==
Syntax	The punctuation/way that code has to be written so that the computer can understand it. Each programming language has its own syntax.
Data Type	This indicates how the data will be stored. The most common data types are integer, string, and float/real.
String	A collection of letters, numbers or characters. (eg, Hello, WR10 1XA)
Integer	A whole number. (eg. 1, 189)
Float/Real	A decimal number, not a whole number. (eg. 3.14, -26.9)
Boolean	1 of 2 values. (eg. True, False, Yes, No)



Python Variable and Type Example



```

x = 10
print("My fav no is " + x)
  
```

Testing your program

Valid - the most obvious or common data that should work

Invalid - data that should definitely fail

Erroneous - data that is the wrong data type

Example – creating a number-guessing game, you will have some code that asks the user to choose a number in a specific range, eg "Choose a number between 1 and 10". To test this unit, you could try a whole range of inputs to see what happens:

3. 4.5. three. -99. 10.00001

For loop	Iteration	While loop
Program: <pre>for i in [5, 4, 3, 2, 1]: print i print 'Blastoff!'</pre>	Output: <pre>5 4 3 2 1 Blastoff!</pre>	Program: <pre>n = 5 while n > 0: print n n = n - 1 print 'Blastoff!' print n</pre>
		Output: <pre>5 4 3 2 1 Blastoff! 0</pre>

Python -> English	
<code>print('hello!')</code>	Prints a value on screen (in this case, hello!)
<code>input('')</code>	Inputs a value into the computer.
<code>x=input('')</code>	Inputs a value and stores it into the variable x.
<code>x=int(input(''))</code>	Inputs a value into x, whilst also making it into an integer.
<code>print(str(x))</code>	Prints the variable x, but converts it into a string first.
<code>if name == "Fred":</code>	Decides whether the variable 'name' has a value which is equal to 'Fred'.
<code>else:</code>	The other option if the conditions for an if statement are not met (eg. name = 'Bob' when it should be Fred)
<code>elif name == "Tim"</code>	elif (short for else if) is for when the first if condition is not met, but you want to specify another option.
<code>#</code>	# is used to make comments in code – any line which starts with a # will be ignored when the program runs.

Helpful websites:

www.computerscience.gcse.guru

w3schools website

learnpython.org

<https://www.programiz.com/python-programming>