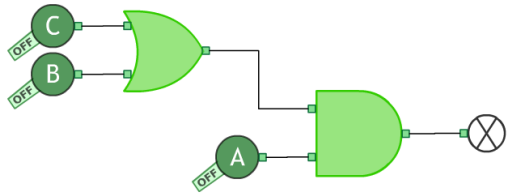
Year 11 Computer Science KO
2.4 Computational logic

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 11 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 11 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

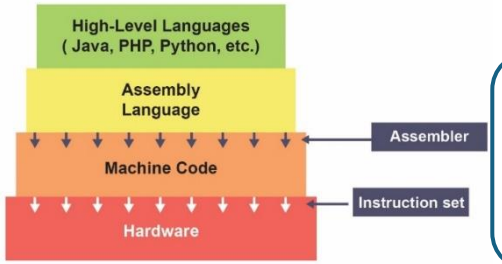
Syntax Checks – Highlights syntax errors

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

Year 11 Computer Science KO – 2.6 Data Representation – Images

Images

- Stored as Bitmap file as pixel
- **Each Pixel** of Image is made up of a 1 or 0.
- Following information about image **is stored in file**:
  - Width of the picture in pixels.
  - Number of bits used for each pixel
  - Colour of each pixel.
- **Image Resolution** = The concentration of pixels in an image
- **Higher Resolution** = More Pixels = Larger File Size
- **Lower Resolution** = Less Pixels = Smaller File Size.

Careers

- Software development
- Photo editor
- UIX designer

Two main types:

BITMAP - The page is divided into an invisible grid and each pixel is assigned a colour

VECTOR

Drawn by following a set of mathematical instructions

- Draw a circle
- radius: 6 pixels
- centre: 10, 10
- line thickness: 1 pixel



Vectors

Vectors are based on mathematical formul and can be scaled infinitely without any loss quality. Every line and shape has a value tha changes when the image expands.

Bitmaps

Bitmaps rely on a series of square blocks called pixels, arranged on a grid. The quality of the images depends on the amount of pixels per square inch. The more pixels, the better the quality.



Metadata: data about data - Certain information must be defined for the bitmap image. E.g. width, height, pixels, colours,

Colour depth

How many bits will be used to store the colour for each pixel in the grid. E.g. 8 bit (1011001) allows 256 different colours.

The greater the colour depth: The more realistic colours, The more data needs to be stored and the larger the file size on disk

Year 11 Computer Science KO – 2.6 Data Representation - Sound

Sound

- The height/amplitude of the sound wave is measured
  - at regular intervals
  - and converted to binary.
- If the interval is smaller
  - More samples taken
  - more data to store
  - larger files
  - the sound reproduced is closer to the original - better quality.

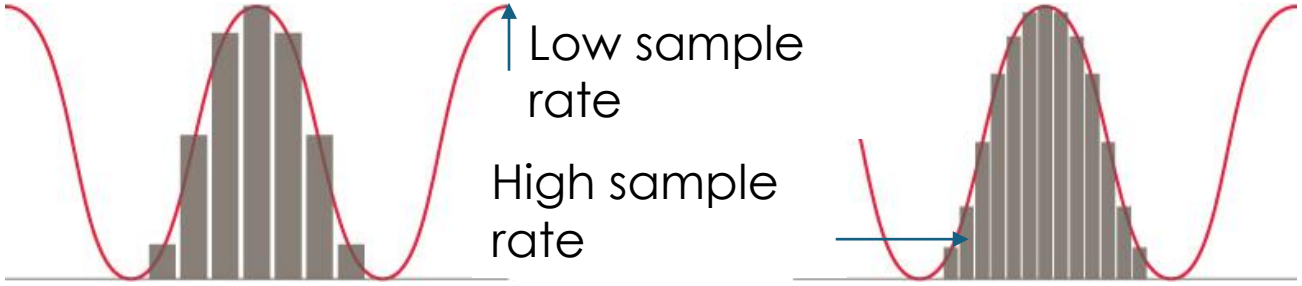
Careers

- Software development
- Sound editor.
- Movie editor.
- Music Software.
- Software Engineering

Digital sound is broken down into **thousands of samples per second** – each of these **samples** is then **stored as binary data.**
The **quality** that the samples are stored with depends on different factors:

- ***Sample Frequency*** **-** The number of audio samples captured every second
- ***Sample Size/ Bit Depth*** **-** Number of bits available for each sample
- ***Bit Rate*** **-** The number of bits TAKEN IN A GIVEN TIME

Sound exists as waves – however as computers only understand binary values this **needs to be converted** into such
Sounds created on a computer exist as digital information that is **encoded as audio** files



Compression: **T**he method computers use to make files smaller by reducing the number of bits (1's and 0's) used to store the information. Can be used for any files including sound and image files
Lossy compression: makes the file smaller by getting rid of bits that aren't really noticeable to the human eye. Once a file has been compressed using a lossy method, any bits that are lost **cannot be recovered**.
Lossless compression: **T**he file is reduced **without losing any quality** so the original file can be restored.
Lossy compression usually has smaller file sizes than lossless

Year 11 Computer Science KO – 2.6 Data Representation - Sound

Bit Rate

- The bit rate of a file tells us how many **bits** of data are processed every second. Bit rates are usually measured in **kilobits per second (kbps)**.

How do you Calculate bit rate?

- The bit rate is calculated using the formula:

Frequency × bit depth × channels = bit rate

- A typical, uncompressed high-quality audio file has a **sample rate** of 44,100 **samples** per second, a bit depth (the number of bits available for each sample) of 16 bits per sample and 2 channels of stereo audio. The bit rate for this file would be:

44,100 samples per second × 16 bits per sample × 2 channels = 1,411,200 bits per second (or 1,411.2 kbps)

A four-minute (240 second) song at this bit rate would create a file size of:

14,411,200 × 240 = 338,688,000 bits (or 40.37 megabytes)

Compression: The method computers use to make files smaller by reducing the number of bits (1's and 0's) used to store the information. Can be used for any files including sound and image files

How is sound stored in Binary?

Audio file is **inputted through a microphone** - broken down into thousands of samples per second,

More samples recorded **per second** the **higher the quality** of the audio file, but the **more memory** it will consume. **Each sound sample** is stored as **Binary Data.**

The **more bits per sample** also increases **the higher the quality** of the audio, on a CD the bit depth is usually 16 bits

Lossy compression: makes the file smaller by getting rid of bits that aren't really noticeable to the human eye. Once a file has been compressed using a lossy method, any bits that are lost **cannot be recovered**. **+ can produce smaller file sizes. - No good if 100% accuracy required e.g. text files**

Lossless compression: The file is reduced **without losing any quality** so the original file can be restored.
+ Original file can be restored without losing any data.
- Not all files can be compressed using lossless compression

Year 11 Computer Science KO – 2.6 Data Representation – Characters

Character Set

is used to describe the possible characters that can be represented in a computer system. E.g A a, 123, @!"£, emoji's

Ascii (American Standard Code for Information Interchange)

- Each character is given a binary code
- Uses **7 Bits** this gives **128 possible characters**
- **Extended Ascii** used **8 bits** – **256 characters** – enough for the English language
- Some codes are reserved for control characters (eg TAB, Carriage Return)

Unicode

- Unicode has a much larger character set
- can represent many more characters/characters from all alphabets
- uses 16 bits
- It **uses 2 bytes** that give us 2^{16} possibilities (65,536).
- This is used universally to represent many more languages than our own

Code	Char	Code	Char	Code	Char	Code	Char	Code	Char	Code	Char
32	[space]	48	0	64	@	80	P	96	`	112	p
33	!	49	1	65	A	81	Q	97	a	113	q
34	"	50	2	66	B	82	R	98	b	114	r
35	#	51	3	67	C	83	S	99	c	115	s
36	$	52	4	68	D	84	T	100	d	116	t
37	%	53	5	69	E	85	U	101	e	117	u
38	&	54	6	70	F	86	V	102	f	118	v
39	'	55	7	71	G	87	W	103	g	119	w
40	(	56	8	72	H	88	X	104	h	120	x
41	)	57	9	73	I	89	Y	105	i	121	y
42	*	58	:	74	J	90	Z	106	j	122	z
43	+	59	;	75	K	91	[	107	k	123	{
44	,	60	<	76	L	92	\	108	l	124	|
45	-	61	=	77	M	93	]	109	m	125	}
46	.	62	>	78	N	94	^	110	n	126	~
47	/	63	?	79	O	95	_	111	o	127	[backspace]

Careers

- Software development
- Programing
- Software Engineering

Year 11 Computer Science KO – 2.3 Robust Programs

Careers
- Software development
- Programing
- Software Engineering

Producing robust programs so that they are defensive against hacks and attacks.

Data Validation: check to ensure that the data entered is sensible and reasonable. It does not check the accuracy of data.
Data Sanitisation: Trims or strips strings, removing unwanted characters to make sure it contains only permitted characters . E.g. Da%ve the % would be removed.
Authentication: The process of verifying the identify of a user or process
Maintainability : Updating code so that it is compatible with current requirements
Comments: To help describe the code and structures in a program using #
Indentation: used to show the programs structure e.g. where selection or iteration or functions are used .

Data Validation techniques

Check digit	the last one or two digits in a code are used to check the other digits are correct	bar code readers in supermarkets use check digits
Format check	checks the data is in the right format	a National Insurance number is in the form LL 99 99 99 L where L is any letter and 9 is any number
Length check	checks the data isn't too short or too long	a password which needs to be six letters long
Lookup table	looks up acceptable values in a table	there are only seven possible days of the week
Presence check	checks that data has been entered into a field	in most databases a key field cannot be left blank
Range check	checks that a value falls within the specified range	number of hours worked must be less than 50 and more than 0
Spell check	looks up words in a dictionary	MS Word uses red lines to underline misspelt words

Anticipating Misuse: Defensive program design will consider and anticipate misuse. Misuse may be in the form of a brute force attack on the program.
E.G Many programs and systems only allow a user to enter a password three or four times before it locks out the system.

Contingency Planning: Once a programmer has anticipated the misuse they can then plan for the these issues. For example: Limiting the number of logon attempts. Ensuring the code is robust in validating the data entered

Year 11 Computer Science KO – 2.3 Robust Programs

Careers
- Software development
- Programing
- Software Engineering

Iterative testing: This happens overtime and is repeated throughout the development of the program

Final/terminal testing: is when the product is released and real end users begin using it. Very often, end users find things wrong with systems that the programmers did not expect.

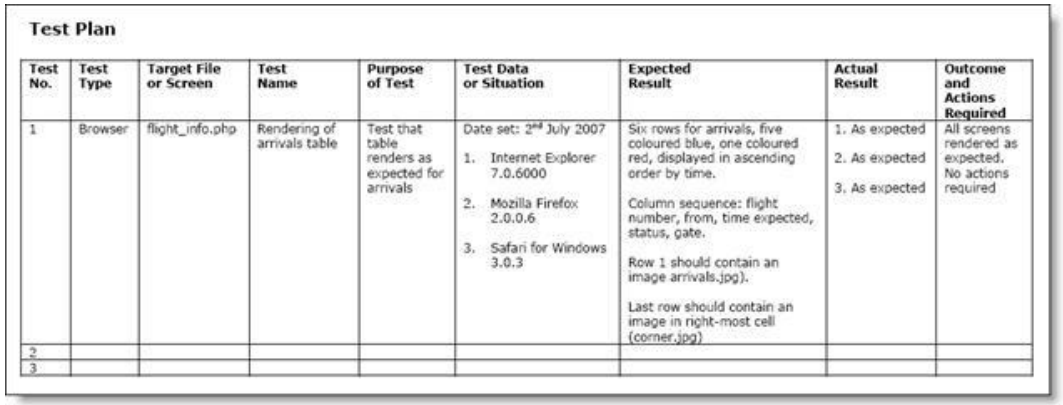
Syntax Errors: An error in the rules/grammar of the language Eg missing colon / spelling mistake

Logic errors: 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.

Test Data: Something which has been specifically identified for use in **tests**, typically of a computer program

Test data – When testing your programs it is important to use of data to test and to create a test plan

- **Valid** – Data that is correct
- **In Range** – The maximum values of the data that could be entered for example for teenagers 13 - 19
- **Out of Range** – Values higher or lower than the expect range, for teenagers greater than 19
- **Null Value** – when no data is entered or left blank to test what happens.
- **Invalid** - incorrect values such as entering 'Dave' in an age field.



Test Plan

Test No.	Test Type	Target File or Screen	Test Name	Purpose of Test	Test Data or Situation	Expected Result	Actual Result	Outcome and Actions Required
1	Browser	flight_info.php	Rendering of arrivals table	Test that table renders as expected for arrivals	Date set: 2nd July 2007 1. Internet Explorer 7.0.6000 2. Mozilla Firefox 2.0.0.6 3. Safari for Windows 3.0.3	Six rows for arrivals, five coloured blue, one coloured red, displayed in ascending order by time. Column sequence: flight number, from, time expected, status, gate. Row 1 should contain an image arrivals.jpg). Last row should contain an image in right-most cell (corner.jpg)	1. As expected 2. As expected 3. As expected	All screens rendered as expected. No actions required
2								
3								