01 Zero & One

Unit 6: Information Design

Programming & AI, 1st secondary (Introduction to ICT)



Engineer Rahim

6-1 Analog and Digital

- **Analog or digital?**
  - **Analog:** a quantity that changes **continuously** and can be measured very finely, such as **mass, time and temperature**
  - **Digital:** the same continuous quantity, but **cut into regular intervals** and written as **numbers**
  - Quick example: a mercury thermometer shows heat by the length of a liquid column (analog)
- **Converting both ways**
  - **A/D conversion:** , also called **digitization**: like the microphone in your phone recording your voice.
  - **D/A conversion:** like headphones turning a song file back into sound your ears can hear.
- **Why digital is better**
  - Data can be **stored** and **copied** without getting worse.
  - It is easy to **modify** and **edit**.
  - It can be **sent** efficiently, and small noise in the signal is easy to correct.
  - Different media (text, images, sound, video) can be **combined** in one place.
- **Binary**: **Binary:** represents numbers with only two digits: **0 and 1**

Analog	A quantity that changes continuously and can be measured very finely, like time or temperature.
Digital	A continuous quantity cut into regular steps and represented as numbers.
A/D conversion	Converting analog data into digital data.
D/A conversion	Converting digital data back into analog data.
Digitization	Another name for A/D conversion: turning analog data into digital data.
Binary	Representing numbers with only two digits, 0 and 1.

6-2 Binary and Amount of Information

- **Bits and bytes**
  - **Bit:** the smallest unit of information, with only two states: **0 or 1** (switch on/off, voltage high/low).
  - **n bits:** represent 2^n values: 1 bit = 2, 2 bits = 4, 3 bits = 8.
  - **Byte: 8 bits**, symbol **B**
  - To find the fewest bits you need, find the smallest n where 2^n is at least the number of things
- **Data units**
  - 1 **KB** = 1,024 B
  - 1 **MB** = 1,024 KB = 2^{20} B
  - 1 **GB** = 1,024 MB = 2^{30} B
  - 1 **TB** = 1,024 GB
- **Decimal and binary**: **Decimal system:** digits 0 to 9
- **Converting between them**

- **Binary → decimal:** from the right, multiply each digit by 1, 2, 4, 8, 16… (2^0, 2^1, 2^2…) and add
- **Decimal → binary:** divide by 2 and write down the remainder each time until the quotient is 1, then read the last quotient and the remainders **from bottom to top**

Bit	The smallest unit of information, with two states: 0 or 1.
Byte	A unit of 8 bits, written B; it can represent 256 values.
KB	Kilobyte: 1,024 bytes.
MB	Megabyte: 1,024 KB.
GB	Gigabyte: 1,024 MB.
TB	Terabyte: 1,024 GB.
Decimal system	Representing numbers with the ten digits 0 to 9.
Binary system	A number system that uses only the digits 0 and 1.

6-3 Hexadecimal

- **What is hexadecimal?**: **Hexadecimal:** uses the digits **0 to 9** and the letters **A to F**, sometimes with a small (16) after the number, like `9A(16)`
- **A table to memorize**
  - A = 10 = `1010` · B = 11 = `1011` · C = 12 = `1100`
  - D = 13 = `1101` · E = 14 = `1110` · F = 15 = `1111`
  - After F comes `10(16)` = 16 = `10000(2)`.
  - From 0 to 9 the digit is the same as in decimal, and in binary it is written in 4 digits (for example 5 = `0101`).
- **How to convert**
  - **Binary → hex:** split the number into groups of **4 digits from the right** and convert each group
  - **Hex → binary:** turn each digit into **4 bits** and put them side by side
  - **Hex → decimal:** go through binary, or directly: left digit × 16 + right digit

Hexadecimal	Representing numbers with the digits 0–9 and the letters A–F.
Hexadecimal number	A number written in base 16, marked with a small (16).
Least significant bit	The rightmost bit, which has the smallest place value.
Binary number	A number written in base 2, marked with a small (2).
Decimal number	An ordinary base-10 number using the digits 0 to 9.

6-4 Digital Representation of Characters

- **Characters are numbers too**
  - **Character code:** a unique number given to each character or symbol.
  - **Character code system:** the table that links each character to its code.
  - **ASCII code:** has only English letters, digits, symbols and computer **control characters**, and each character takes **1 byte**
  - **Unicode:** a standard that brings the characters of **all the world's languages** into one system, with forms like **UTF-8** and **UTF-16**.
- **Reading the ASCII table**: The table has the **upper 4 bits** (column) and the **lower 4 bits** (row)
- **Encoding and decoding**
  - **Encoding:** turning text into character codes

- **Character corruption:** text shows strange symbols when the encoding system differs from the decoding system.
- **Font:** the **shape** data of a character that matches its code, such as Serif and Sans-serif.
- To show a character on a screen or printer you need two things: **the character code + a font**.

- **Quick maths**: 1 byte = 2^8 = **256** possible characters, and 2 bytes = 2^{16} = **65,536** characters

Character code	A unique number assigned to each character or symbol.
Character code system	A table that links characters to their character codes.
ASCII code	A code system for English letters, digits, symbols and control characters, 1 byte each.
Unicode	A standard that combines the characters of all the world's languages (UTF-8, UTF-16).
Encoding	Representing text with character codes.
Decoding	Turning character codes back into text.
Character corruption	Strange symbols that appear when encoding and decoding systems do not match.
Font	The shape data of characters that matches their character codes.

6-5 Numerical Calculations [1]

- **Binary addition**
  - `0 + 0 = 0` · `0 + 1 = 1`
  - `1 + 1 = 10` (write 0, carry 1)
  - `1 + 1 + 1 = 11` (write 1, carry 1)
- **Binary subtraction**
  - `0 - 0 = 0` · `1 - 0 = 1` · `1 - 1 = 0`
  - `10 - 1 = 1`
- **Check your answer**: The easiest check: convert the numbers to decimal, calculate, then convert the result to binary and compare.

Binary addition	Adding digit by digit, carrying 1 when a digit's sum reaches 2.
Binary subtraction	Subtracting digit by digit, borrowing 2 from the higher digit when needed.
Carry	Moving 1 to the next higher digit when a digit's sum reaches the base.
Borrow	Taking 1 from the higher digit, worth 2 in the current binary digit.
Digit	One position holding a single figure inside a number.

6-6 Numerical Calculations [2]

- **What is a complement?**: **Complement:** the smallest number that, added to a given number, causes a **carry** to the next higher digit
- **In decimal**
  - **9's complement:** subtract each digit from 9
  - **10's complement:** the 9's complement + 1
  - Two-digit example: the 10's complement of 64 = **36** (because 64 + 36 = 100).
- **In binary**

- **One's complement:** flip every bit (0 → 1 and 1 → 0)
- **Two's complement:** (the binary complement): flip every bit **and then add 1**
- Check: `0110 + 1010 = 10000`

- **Subtracting with complements**
  - 1) Find the **complement** of the subtrahend.
  - 2) **Add** it to the other number.
  - 3) **Ignore the extra leading digit**
  - Decimal example: 9 − 4 → the 10's complement of 4 = 6 → 9 + 6 = 15 → drop the 1 → **5**.
  - Binary example: `1011 - 0101` → two's complement of 0101 = `1011` → `1011 + 1011 = 10110` → drop the 1 → `0110` (11 − 5 = 6).

Complement	The smallest number that, added to a number, causes a carry to the next higher digit.
9's complement	Subtracting each decimal digit from 9.
10's complement	The 9's complement plus 1.
One's complement	Flipping every bit of a binary number.
Two's complement	Flipping every bit and then adding 1.
Subtrahend	The number that is being subtracted.

6-7 Digitalization of Sound

- **Sound is a wave**
  - **Sound:** a vibration travelling through the air, so it is **analog** data shaped like a wave
  - **Frequency:** the number of waves in one second, measured in **hertz (Hz)**.
  - **Period:** the time one wave takes, measured in **seconds**.
- **PCM — 3 steps in order**
  - **1) Sampling:** divide the **time axis** (horizontal) at regular intervals and take the wave height at each point (**sample points**)
  - **2) Quantization:** divide the **voltage axis** (vertical) into regular levels and round each sample to the nearest level
  - **3) Encoding:** write each value in binary.
- **Quality versus size**
  - The higher the **sampling frequency** and **bit depth**, the closer the sound is to the original wave and the better the quality, but the **data gets bigger**.
  - **Sampling theorem:** if the sampling frequency is **more than twice** the highest frequency in the original wave, the original wave can be rebuilt accurately
  - **Channels:** one channel = **monaural**, two channels = **stereo**.
- **The sound size formula**: **Data (bits) per second = sampling frequency × bit depth × number of channels:** , then multiply by the number of seconds and divide by 8 to get bytes.

Frequency	The number of waves in one second, measured in hertz (Hz).
Period	The time one wave takes, measured in seconds.
PCM	Pulse Code Modulation: a method that turns analog sound into binary code.
Sampling	Dividing the time axis at regular intervals and reading the wave height.
Sampling frequency	The number of samples taken per second.

Quantization	Rounding each sample to the nearest level on the voltage axis.
Quantization bit depth	The number of bits that sets how many levels are used (n bits = 2^n levels).
Encoding	Writing the quantized values in binary.
Sampling theorem	Sampling at more than twice the highest frequency lets the original wave be rebuilt.
Stereo	Playback with two different signals (channels); monaural uses one.

6-8 Digitization of Images

- **A digital image = pixels**
  - **Pixel:** the smallest unit of an image
  - **1) Sampling:** split the image into pixels and take the brightness of each one
  - **2) Quantization:** turn each pixel's brightness into a number from a set of levels
  - **3) Encoding:** write the values in binary.
- **Image size**: **Image data (bits) = number of pixels (width × height) × color bits per pixel:** , then ÷ 8 to get bytes.
- **Raster or vector?**
  - **Raster format:** the image is a grid of pixels
  - **Vector format:** stores the **coordinates of points** and the angles and thickness of the lines joining them, so it stays smooth at any size
- **Colors**
  - **Three primary colors of light:** red, green and blue (RGB)
  - **Three primary colors of pigment:** cyan, magenta and yellow (CMY)
  - **24-bit full color:** each of the 3 colors has 256 levels (8 bits), so 8 × 3 = 24 bits = 2^{24} = **16,777,216** colors.
  - **Color depth:** the number of bits used for each pixel

Pixel	The smallest unit that makes up an image.
Resolution	How fine the pixels are when sampling, measured in dpi.
Gradation	The number of levels of color intensity per pixel.
dpi	Dots per inch, the unit of resolution.
Raster format	Representing an image as a grid of pixels.
Vector format	Representing an image by point coordinates and the angles and thickness of lines.
Jaggies	The jagged edges seen when a raster image is enlarged.
Additive color mixing	Mixing the primary colors of light (RGB), used in screens.
Subtractive color mixing	Mixing the primary colors of pigment (CMY), used in printers.
24-bit full color	8 bits for each of red, green and blue: 24 bits, about 16.7 million colors.
Color depth	The number of bits used for each pixel; it sets how many colors are possible and the file size.

6-9 Digital Representation and Compression Technology for Videos

- **How does video work?**

- **Video:** still images shown one after another quickly, so the eye senses motion because of the **afterimage phenomenon**
- **Frame:** each still image inside a video.
- **Frame rate:** the number of frames per second, measured in **fps**

- **Video size**: **Video size = frame size (B) × frame rate (fps) × time (seconds):** .
- **Data compression**
  - **Compression:** making data as small as possible while keeping its content
  - **Compression ratio % = (size after compression ÷ original size) × 100:**
  - **Lossless compression:** restores the original data **exactly**
  - **Lossy compression:** does not restore the original exactly, but people do not notice the difference
- **Common lossless methods**
  - **Run-length encoding:** replaces repeated symbols in a row with the symbol + how many times it repeats
  - In a black-and-white grid: read the pixels row by row from the top left, B for black and W for white
  - **Huffman coding:** gives a **shorter code** to the characters that **appear most**, and a longer code to rare ones.

Video	A series of still images shown in quick succession to create the illusion of motion.
Afterimage phenomenon	A feature of human vision that keeps an image briefly, making still frames look like motion.
Frame	One still image of the series that makes up a video.
Frame rate	The number of frames shown per second, measured in fps.
Compression	Reducing the amount of data as much as possible while keeping its content.
Decompression	Restoring compressed data to its original state.
Compression ratio	Size after compression ÷ original size × 100.
Lossless compression	Compression that lets the original data be restored exactly.
Lossy compression	Compression where the original data cannot be fully restored.
Run-length encoding	Replacing a run of identical symbols with the symbol and its count.
Huffman coding	Giving shorter bit codes to characters that appear more often.

6-10 Information Design

- **What is information design?**: **Information design:** organizing information and presenting it creatively for its purpose, so the message reaches the intended audience correctly.
- **3 methods of information design**
  - **Abstraction:** take only what matters from a large amount of data and present it simply
  - **Visualization:** show information visually so it is understood faster, like **tables and graphs**.
  - **Structuring:** arrange information by relationships, levels and order, like **hierarchical menus** on websites and a mall floor map.
- **Easy to understand and use**
  - **User interface (UI):** the system that passes information between people and devices (voice, touch screen, keyboard).
  - **CUI:** you type commands on a keyboard
  - **User experience (UX):** the feeling and experience a user takes away from a product or service.
  - **Affordance:** the actions that can be done with a thing (a chair can be sat on)

- **Usability:** how easy and understandable something is for its users

- **The LATCH method for organizing information**
  - **L
  - **A
  - **T
  - **C
  - **H

- **Data design**
  - **Data design:** the stage where we decide what data a system or database needs and how it is organized
  - **First step:** analyze and understand the app's **data requirements**, before choosing a programming language or writing anything.
  - **Rules of good design:** define **data types** accurately (number, text, date...), arrange the data in tables, link related fields, and avoid **unnecessary data duplication**.
  - **Why it matters:** good design reduces **inconsistency and conflict** in the data, keeps the system fast, and makes later changes and development easy

Information design	Organizing and presenting information for its purpose so the message reaches its audience correctly.
Abstraction	Picking out only the needed information from a large amount of data and conveying it simply.
Pictogram	A picture symbol that gives information without using words.
Visualization	Showing information visually, for example as tables and graphs.
Structuring	Arranging information by relationships, levels and order.
CUI	A user interface operated by typing commands on a keyboard.
GUI	A user interface operated intuitively with icons and buttons.
User experience (UX)	The experience and feeling a user gets from a product or service.
Signifier	A cue that prompts the user to take an action.
Universal design	Design made so that everyone can use it without difficulty.
Data design	Organizing and arranging the data a system needs so it is easy to access and process.
Data types	The kind of value each field holds, such as number, text or date, defined accurately during design.
Data duplication	Storing the same data more than once without need; good data design avoids it.



Study the whole course free

Every lesson, questions with answers, a ministry-style bank and a timed mock exam

zerooneacademy.com

Engineer Rahim