IDX G10 CS H
Study Guide Issue 1
By Gavin Yu 10-4, Edited by Ethan Qin

NOTE: This is an official document by Indexademics. Unless otherwise stated, this document may not be accredited to individuals or groups other than the club IDX, nor should this document be distributed, sold, or modified for personal use in any way.

Introduction to Computer Science

- **Definition:** Programmable electronic device → receives input, processes/stores data, produces output.
- **Key functions:**
  - Input
  - Processing
  - Storage
  - Output
- **CS** is the study of **information, computation, and algorithms**, and how they are implemented via hardware/software.
- **Covers:**
  - Theoretical foundations
  - Algorithms & logic
  - Data structures
  - Programming
  - Networks
  - Artificial Intelligence (AI)

The Big Picture

1. Components of a Computer

- Hardware: Physical components (CPU, RAM, HDD, I/O devices)
- Software: Programs and OS that tell hardware what to do
    - System Software: eg Operating System, Firmware, IDE, Database management, etc
    - Application Software: Special programming to complete special tasks
- Network: Connects devices for data sharing & communication

2. Types of Computers (by Computational Power)

Type	Description	Example Use
PC (Personal Computer)	Least powerful, most common	Home, office
Midrange/Server	Serve multiple users, host databases	Web servers
Mainframe	Very large, high storage/speed	Banks, enterprises
Supercomputer	Highest performance	Scientific research, AI

3. Historical Milestones

Machine	Year	Achievement
Atanasoff-Berry Computer	1940	First computer
ENIAC	1946	First general-purpose electronic computer
Manchester Baby	1948	First stored-program computer

4. AI Milestones

- Deep Blue vs Kasparov (1996–1997): First chess AI victory
- AlphaGo (2016): Defeated world champion in Go using ML + neural networks
- Supercomputers today: Thousands of CPUs/GPUs (Top500.org)

Data Representation

1. Data vs. Information

- Data: Symbols that represent facts, numbers, text, images, etc.
- Information: Processed data presented in a meaningful form understandable by humans

2. Data Representation

- How data is stored, processed, and transmitted in computers.
- Digital vs Analog
    - Digital: Discrete (0,1) – used by computers
    - Analog: Continuous (e.g., sound wave)

The Binary System

1. Basic Concepts

- Binary digits (bits): 0 or 1, stored as electrical voltage in transistors
- Byte: 8 bits
    - 1KB = 1024B
    - 1MB = 1024KB
    - 1GB = 1024MB
    - etc.
- All data are stored as bits

2. Number Systems

System	Base	Digits
Binary	2	0, 1
Octal	8	0–7
Decimal	10	0–9
Hexadecimal	16	0–9, A–F

3. Base Conversions

(a) Decimal → Binary

- Divide by 2 repeatedly, record remainders (bottom-up).

  Example: 23 → 10111_2

(b) Binary → Decimal

- Multiply each bit by 2^n and sum.
  Example: $10111_2 = 16+4+2+1 = 23_{10}$

(c) Fractional parts

- Multiply fractional part by 2 repeatedly.
  Example: $0.25_{10} = 0.01_2$
- $0.63_{10} \rightarrow$ approx. 0.10100011_2

(d) Binary → Hexadecimal

- Divide binary number into groups of 4, adding 0s at the start if needed.

4. Binary Arithmetic

- Addition Rules:
  0+0=0 | 0+1=1 | 1+0=1 | 1+1=10
- Subtraction Rules:
  0−0=0 | 1−0=1 | 1−1=0 | 0−1=1 (borrow)
- Carry/Borrow similar to decimal arithmetic

Negative Numbers in Binary

1. True Form (Sign-Magnitude)

- MSB (Most Significant Bit) = sign bit
  - 0 → positive
  - 1 → negative
- Example:
  +5 → 00000101
  −5 → 10000101

2. 1's Complement (Not suitable for calculation)

- Flip all bits of a binary number.
  +5 (00000101) → −5 (11111010)
- Two zeros exist: +0 (00000000), −0 (11111111)

3. 2's Complement (Suitable for calculation)

- Take 1's complement + 1
  -5 (true form) → 10000101
  1's → 11111010
  2's → 11111011 (−5)

- Only one zero (00000000)
- Most used in computers

Representing Characters

Code	Description	Notes
ASCII	7-bit, 128 chars	English letters, digits, symbols
ANSI	8-bit, 256 chars	Windows extension
Unicode	Up to 4 bytes	Universal, supports all languages
UTF-8/16/32	Encoding methods	UTF-8 variable length (1–4 bytes)

Image Representation

1. Representing Color

- Primary colors: Red, Green, Blue (RGB)
- Color depth: Bits per pixel (bpp)
    - 8-bit → 256 colors (2^8)
    - 24-bit (True Color) → 16.7 million colors

2. Digital Images

- Pixel: Smallest unit of an image
- Resolution: # of pixels (width × height)
- File formats: BMP, JPG, GIF
- Raster Graphics: Image stored pixel-by-pixel

3. Monochrome Example

00111100

01100110

11100111…

Each bit = one pixel (black/white)

4. Steganography

- Encryption of information into a picture.
- Example: Hide info inside least significant bits (LSB) of image pixels.

5. HDR

- Combine multiple exposures → better contrast and details

6. Codecs

- Used in photos and videos
- Each format has its own codec: format = container (contains data) + codec

Sound Representation

- Sampling: Take amplitude samples at fixed intervals
- Sample Rate: Samples per second (Hz)
- Quantization: Round each sample to nearest binary value
- Higher rate/bit depth → better quality, larger file

Data Compression

1. Why Compression?

- Reduce storage, transmission time

2. Types

Type	Description	Example
Lossless	No data loss	Run-Length Encoding (RLE), Huffman Coding
Lossy	Some data discarded	JPEG, MP3, MP4

3. Run-Length Encoding (RLE)

- Store consecutive same data as count + value
    - e.g., 4W3G4W3G2W

4. Huffman Encoding

- Variable-length codes based on frequency
- Common symbols → shorter codes
- Example (freqs): Q(0.12), W(0.34), E(0.28), R(0.09), T(0.17)
- Encoding table is stored in metadata

5. Compression Ratio

= (Original Size) / (Compressed Size)