## Components/Processor Registers

| | |
|---|---|
| Mother board | The central interface for all the components of a PC. Everything connects to the motherboard via slots, wires, readouts and connectors. |

## Components/Processor Registers (cont)

| | |
|---|---|
| Processor (CPU) | A combination of registers than manipulate data between the registers. The speed of a processor is measured in the number of instructions it can complete per second (Hz). Modern computers speed is measured in GHz. |

By **0llieC** (0llieC)
cheatography.com/0lliec/

Published 20th February, 2019.
Last updated 12th June, 2017.

## Components/Processor Registers (cont)

| | |
|---|---|
| Secondary Storage | Used to store programs and data. It can be partitioned to allow for dual-booting multiple operating systems. |

## Components/Processor Registers (cont)

## Components/Processor Registers (cont)

**Processor Structure**

By **0llieC** (0llieC)
cheatography.com/0lliec/

Published 20th February, 2019.
Last updated 12th June, 2017.

## Components/Processor Registers (cont)

## Components/Processor Registers (cont)

| | |
|---|---|
| Arithmetic Logic Unit (ALU) | The ALU carries out arithmetic calculations and comparisons. The result of any calculation is sent to the Accumulator |

## Components/Processor Registers (cont)

| | |
|---|---|
| Control Unit (CU) | The CU controls the operation of the hardware, inc. input and output devices, it controls the Fetch-Decode-Execute cycle. |

By **0llieC** (0llieC)
cheatography.com/0lliec/

Published 20th February, 2019.
Last updated 12th June, 2017.

## Components/Processor Registers (cont)

| | |
|---|---|
| Clock | The clock is the part that regulates the cycle of the CPU. It provides a regular pulse of high voltage then low voltage. This high-low transition is a cycle, each cycle is an instruction |

## Components/Processor Registers (cont)

| | |
|---|---|
| Program Counter (PC) | This register holds the address of the next instruction to be executed, the PC is automatically implemented to the next instruction, unless the previous instruction was a jump. |

## Components/Processor Registers (cont)

| | |
|---|---|
| Memory Buffer Register (MBR) | Values fetched from memory are sent to MBR. |

By **0llieC** (0llieC)
cheatography.com/0lliec/

Published 20th February, 2019.
Last updated 12th June, 2017.

## Components/Processor Registers (cont)

| | |
|---|---|
| Memory Address Register (MAR) | The location in memory of the current instruction/data being fetched. |

## Components/Processor Registers (cont)

| | |
|---|---|
| Current Instruction Register (CIR) | The instruction currently being executed/decoded |

## Components/Processor Registers (cont)

| | |
|---|---|
| Data Bus | Carries the data between memory and the MBR |

By **0llieC** (0llieC)
cheatography.com/0lliec/

Published 20th February, 2019.
Last updated 12th June, 2017.

## Components/Processor Registers (cont)

| | |
|---|---|
| Address Bus | Carries the memory location of the instructions/data being received. |

## Components/Processor Registers (cont)

| | |
|---|---|
| Control Bus | A bus with 2 states, set or enable, which govern if the data bus is reading or writing to memory |

## Components/Processor Registers (cont)

By **0llieC** (0llieC)
cheatography.com/0lliec/

Published 20th February, 2019.
Last updated 12th June, 2017.

## Components/Processor Registers (cont)

**Fetch Decode Execute**

## Components/Processor Registers (cont)

## Components/Processor Registers (cont)

**Fetch**

The PC contents are copied to the MAR

Instruction at address in MAR → MBR

MBR → CIR

↓

**Decode**

Instruction is decoded into:

1. Operand → The data to preform an instruction on

2. Op-Code → The instruction

↓

**Execute**

Instruction executed

If data is being committed to memory, its held in the MBR

Cycle repeats until stop instruction

By **0llieC** (0llieC)
cheatography.com/0lliec/

Published 20th February, 2019.
Last updated 12th June, 2017.

## Components/Processor Registers (cont)

By **0llieC** (0llieC)
cheatography.com/0lliec/

## Components/Processor Registers (cont)

### Factors affecting Processor Performance

Published 20th February, 2019.
Last updated 12th June, 2017.


## Components/Processor Registers (cont)

## Components/Processor Registers (cont)

| | |
|---|---|
| Multiple Cores | The increase in number of cores, allows for a greater throughput of data. If the software is threaded - can use multiple cores - it will divide up tasks to the different cores. **However**, it must be coded in, else it will use the single core. |

## Components/Processor Registers (cont)

| | |
|---|---|
| Cache Size | Cache is a small amount of **very** fast memory. Repeatedly used instructions and data is stored in the cache for quick access. The bigger the cache, the more can be stored on it thus reducing processing time |

## Components/Processor Registers (cont)

| | |
|---|---|
| Clock Speed | The clock regulates the instruction execution rate. The faster the clock, the more cycles completed per second. |

By **0llieC** (0llieC)
cheatography.com/0lliec/

Published 20th February, 2019.
Last updated 12th June, 2017.

## Components/Processor Registers (cont)

| | |
|---|---|
| Pipelining | Where the stages of the F-D-E cycle are 'stacked' so that they can be processed at the same time. While one instruction is being fetched, the previous is being decoded. This may not necessarily increase processing time but throughput is increased.<br><br>**Issues**<br>• If an instruction requires the result of a previous instruction, the CPU will remain dormant → leading to 'bubbles'/pipeline stalls in the pipeline.<br>• Jumps → lead to the pipeline having to be flushed due to the change in instructions<br><br>**Hyper-Threading** → Where the CPU is intelligent enough to fill the bubbles caused by pipeline stalls with other non-dependant instructions from separate threads. |

## Components/Processor Registers (cont)

## Components/Processor Registers (cont)

**Processor Architecture**

## Components/Processor Registers (cont)

## Components/Processor Registers (cont)

| | |
|---|---|
| Von-Neuman | The Von-Neumann architecture is commonly used in most PCs. It stores both programs and data in the same memory. Using the F-D-E cycle, it carries out a single instruction at a time.<br><br>**Pros**<br>+ More Robust than Harvard (older)<br>+ Single Storage<br><br>**Cons**<br>- Each Instruction takes 2 cycles (fetch/decode and execute)<br>- Cannot implement pipelining |

## Components/Processor Registers (cont)

| | |
|---|---|
| Harvard | The Harvard architecture stores programs and data in **separate** memory and uses the control unit at the centre of the structure. Generally used in embedded systems.<br><br>**Pros**<br>+ Can complete an instruction in a single clock cycle (assuming pipelining is used)<br>+ Modern<br>+ Can implement pipelining<br><br>**Cons**<br>- Separate Memory |

By **0llieC** (0llieC)
cheatography.com/0lliec/

Published 20th February, 2019.
Last updated 12th June, 2017.

## Components/Processor Registers (cont)

By **0llieC** (0llieC)
cheatography.com/0lliec/

## Components/Processor Registers (cont)

### RISC/CISC

Published 20th February, 2019.
Last updated 12th June, 2017.


## Components/Processor Registers (cont)

## Components/Processor Registers (cont)

CISC **CISC (Complex Instruction Set Computing)**

• One instruction can complete an entire sequence - more complex
• Higher Power Consumption
• Powerful
• Generates more heat

## Components/Processor Registers (cont)

RISC **RISC (Reduced Instruction Set Computing)**

• Only one value fetched/stored per instruction cycle
• Less Power Required
• Used in smaller devices (Smartphones)
• Generates less heat - requires less cooling methods

## Components/Processor Registers (cont)

## Components/Processor Registers (cont)

**Flynn's Taxonomy**

## Components/Processor Registers (cont)

## Components/Processor Registers (cont)

SISD **Single Instruction, Single Data-stream**
Single Core CPUs

**NB**:
• No parallelism
• Single CU, fetches single instruction

By **0llieC** (0llieC)
cheatography.com/0lliec/

Published 20th February, 2019.
Last updated 12th June, 2017.

## Components/Processor Registers (cont)

SIMD **Single Instruction, Multiple Data-streams**
GPUs

**NB**:
• One instruction preformed on many data-streams
• Naturally parallelised operations
• Examples: Fractal Rendering, Graphics Processing (hence GPUs) - each pixel is independant

## Components/Processor Registers (cont)

MIMD **Multiple Instructions, Multiple Data-streams**
Multi-core CPUs

**NB**:
• Multiple autonomous processors simultaneously executing different instructions on different data
• Uses either one shared memory space or a distributed memory space.

## Software Generation

Specific Application — A piece of software that has a specific purpose, such as order entry, payroll, stock management etc. It may be **Bespoke** (made to order) or **Off-the-shelf** (designed to be used in a variety of situations).

By **0llieC** (0llieC)
cheatography.com/0lliec/

Published 20th February, 2019.
Last updated 12th June, 2017.

## Software Generation (cont)

**General Purpose Application**

An application that allows the user to produce their own solution to a problem. Most are sold as a package/a license.

**Examples**

- Word Processing
- Desktop Publishing
- Spreadsheets
- Database Management
- CAD/CAM
- Presentation Software

**Specific Examples**

- Microsoft Office
- Adobe Suite

## Software Generation (cont)

**Open Source vs. Closed Source**

**Open Source** → Source code is readable to anybody and freely modifiable.

**Closed Source** → Executable only, source code is kept hidden.

**Pros of OS**

- Free (usually)
- Community Coding/Bug Fixing - Usually faster than any closed-source
- Customisable
- Freedom to do what you like

**Pros of CS**

- Professional Development
- Lower security risks
- Well documented and customer support

## Software Generation (cont)

**Translator Software**

Software that convert one programming language into another. There are 3 catagories: Compilers, Interpreters and Assemblers.

**Assemblers** → Convert Assembly into machine specific machine code. Assembly language consists of mnemonics that represent different instructions. It is converted to binary (machine code)

**Interpreters** → Checks and executes code line by line.

**Compiler** → Checks all the codes syntax, semantics and logic and then converts the code into Object code (usually machine code or similar low-level). The compiled code is usually what is distributed.

## Software Generation (cont)

By **0llieC** (0llieC)
cheatography.com/0lliec/

## Software Generation (cont)

### Stages of Compilation

Published 20th February, 2019.
Last updated 12th June, 2017.


## Software Generation (cont)

## Software Generation (cont)

**Lexical Analysis**

↓

## Software Generation (cont)

Firstly the code is striped of anything uneeded such as comments and redundant whitespace

The code is then divided into Lexemes (the smallest 'unit' of code).

Tokens are then assigned to each lexeme indicating what it is. Some token examples:
• Identifiers - for variables, subroutines, classes etc.
• Keywords - new, if, for, while etc.
• Operators - +, - , / , == etc.
• Literals - fixed numbers and strings
• Symbols - {}, (), ; etc.

**Errors** are caused when a lexeme cannot be assigned a token

## Software Generation (cont)

**Syntax Analysis**

↓

## Software Generation (cont)

The stream of tokens generated is then analysed to check they match the rules of the language. Tree data structures are often used in this process.

An example of valid syntax would be:

String word = "Hello, World!";
Datatype Literal Operator StringLiteral Symbol

**Errors** occur when a series of tokens cannot be matched to a rule, such as multiple datatypes.

## Software Generation (cont)

**Semmantic Analysis**



## Software Generation (cont)

The stage where code is checked for logical errors. For example:

- Datatype mismatch - assigning a String to an int
- Undeclared Variables, or out of scope variables
- Multiple variable declarations
- Array out of bounds with an integer literal.

**Errors** occur when one of the rules are broken. **NB** Not all semantic errors can be caught during compilation. For example accessing an array with a integer variable is logically fine, but the integer value may be out of bounds causing a run-time error.

By **0llieC** (0llieC)
cheatography.com/0lliec/

Published 20th February, 2019.
Last updated 12th June, 2017.

## Software Generation (cont)

**Intermediate Code Generation/Optimisation**

↓

## Software Generation (cont)

The code is then converted into intermediate code (Java to Java bytecode, where it remains until use - not all languages do this)

**Intermediate code is machine *independent***

The code is then optimised, so that it runs faster and requires less resources, but still having the same output.

## Software Generation (cont)

**Machine Code Generation/Optimisation**

↓

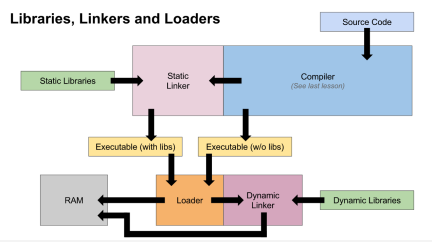
By **0llieC** (0llieC)
cheatography.com/0lliec/

Published 20th February, 2019.
Last updated 12th June, 2017.

## Software Generation (cont)

The final stage of compilation is the conversion to machine code. **This process has to be repeated for each processor as it is machine dependant**. Specific optimisations are also done on the separate processors as code that works well with one instruction set may not work as well with another.

## Libraries, Linkers and Loaders




**Library** → Generic name for a collection of programs used in development. Some languages have native ones. Saves time as the developers don't have to create their own code.
**Linker** → Combines object and library files
**Loader** → Loads the object code into memory to be executed

## Testing Strategies

| | |
|---|---|
| Black Box (Alpha) | Testing that examines the functionality of a application, without looking at its internal code/structures. |

## Testing Strategies (cont)

| | |
|---|---|
| White Box | Tests the internal structure/workings of a application rather than its functionality (opposite of black box) |

## Testing Strategies (cont)

| | |
|---|---|
| Top Down | Testing of modules and sections of code that aren't yet implement. Testing the behaviour between modules. |

## Testing Strategies (cont)

| | |
|---|---|
| Top Down | Testing of modules and sections of code that aren't yet implement. Testing the behaviour between modules. |

By **0llieC** (0llieC)
cheatography.com/0lliec/

Published 20th February, 2019.
Last updated 12th June, 2017.

## Testing Strategies (cont)

| | |
|---|---|
| Bottom Up | Testing each part of the application individually then testing the parts that rely on the section/module. |

## Testing Strategies (cont)

| | |
|---|---|
| Usability (Beta) | Testing how easy a system is to use by testing it with real users. It shows how somebody without a working knowledge of the application would use the system and any problems they might find. |

## Testing Strategies (cont)

| | |
|---|---|
| Test Data | A range of test data must be used to properly test a system. It should include:<br>• Normal Data<br>• Boundary Data<br>• Standard Incorrect Data - incorrect data that could easily be entered<br>• Standard invalid data - e.g. text into numeric fields<br>• Extreme data - data that would never be entered normally, used to test the limits of a system |

By **0llieC** (0llieC)
cheatography.com/0lliec/

Published 20th February, 2019.
Last updated 12th June, 2017.

## OOP

| | |
|---|---|
| Class | A 'blueprint', a combination of attributes and methods that create an object. |

## OOP (cont)

| | |
|---|---|
| Object | An instance of a class. |

## OOP (cont)

| | |
|---|---|
| Encapsulation | Where attributes and methods are wrapped in their objects. Access modifiers control how the methods and the attributes can be accessed, whether that be by any class, or only within its own class. |

By **0llieC** (0llieC)
cheatography.com/0lliec/

Published 20th February, 2019.
Last updated 12th June, 2017.

## OOP (cont)

| | |
|---|---|
| Inheritance | A relationship among classes where a child class shares methods/attributes with its parent class. The child classes can also have their own independent attributes/methods but all child nodes share the ones inherited from the parent. |

## OOP (cont)

| | |
|---|---|
| Abstract Class | A class which contains attributes and methods like a normal class, but the class itself cannot be instantiated. An example is an Animal, you can write an abstract class, but you cannot create just an Animal. |

## OOP (cont)

| | |
|---|---|
| Polymorphism | A feature of a programming language that allows routines to use variables of different types at different times. For example, overloading constructors which behave differently depending on their parameters |

By **0llieC** (0llieC)
cheatography.com/0lliec/

Published 20th February, 2019.
Last updated 12th June, 2017.

## Web Technologies

| | |
|---|---|
| HTML (Hypertext Markup Language) | The standard language for displaying webpages. A HTML document starts with <!DOCTYPE html> <html><br><br>It consists of tags which are opened and closed <tag></tag>. Each document has a head and a body. |

## Web Technologies (cont)

<h1>{{ml}}<h2>...

## Web Technologies (cont)

<a href="http://www.google.com">Link Text</a>

By **0llieC** (0llieC)
cheatography.com/0lliec/

Published 20th February, 2019.
Last updated 12th June, 2017.

## Web Technologies (cont)



## Web Technologies (cont)



## Web Technologies (cont)

| | |
|---|---|
| CSS (Cascading Style Sheet) | The standard way to style webpages, whether internal or externalised. Externalised stylesheets allow developers to keep design and content completely separate. |

By **0llieC** (0llieC)
cheatography.com/0lliec/

Published 20th February, 2019.
Last updated 12th June, 2017.

## Web Technologies (cont)

| | |
|---|---|
| JS (Java Script) | JS is an interpreted code that adds interactivity to websites. It works on virtually all hardware and is used on nearly all websites. For example there are currently 300,000 JS repos on Github, Java has 200k |

## Web Technologies (cont)

| | |
|---|---|
| Search Engine Indexing | A search engine searches through webpages, for certain keywords and phrases. Problem Indexes are used, when a new document (webpage) is added, the words/phrases are tokenised, and added to the list. |

## Web Technologies (cont)

| | |
|---|---|
| PageRank Algorithm | Google's algorithm that calculates the weighting of webpages. All pages have an initial rank, but for each link, it gives a certain amount to the webpage linked. Other algorithms are also used to give pages different rankings depending on what the user is searching for. |

By **0llieC** (0llieC)
cheatography.com/0lliec/

Published 20th February, 2019.
Last updated 12th June, 2017.

## Web Technologies (cont)

| | |
|---|---|
| Client-Side Processing | Processing preformed in the browser, usually JS. This allows user entered data to be checked before sending it to the server, which reduces the load on the server. For example, ensuring an email has an @, or a password is a certain number of characters. Anybody can view the code for client-side processing, so its best for just verification |

## Web Technologies (cont)

| | |
|---|---|
| Server-Side Processing | Processing performed on the server. The code is only viewable to people with access to the server-files. It processes requests and serves a webpage based on the requests |

## L.O.R. (*) Questions

| | |
|---|---|
| Data Protection Act | ***8 Principles***:<br>1. Personal data must be obtained **lawfully** and **fairly**<br>2. P.D. must be held for a **specified** purpose<br>3. P.D. must be **adequate, relevent** and **not excessive**<br>4. P.D. must be kept **up-to-date** and **accurate**<br>5. P.D. must not be kept **longer than necessary**<br>6. P.D. must be processed in accordance with **data subjects** rights<br>7. P.D. must be kept **securely**<br>8. P.D. must not be **transferred outside the EU without permission** |

## L.O.R. (*) Questions (cont)

| | |
|---|---|
| Computer Misuse Act | **Level 1: Unauthorised Access**<br>• Accessing secure parts of a computer, that they are unauthorised to access<br>• In organisations, accessing secure parts that are beyond your rights.<br><br>**Level 2: Unauthorised Access with intent to commit a Crime**<br>• Level 1 + intent to commit another crime.<br><br>**Level 3: Unauthorised Modification**<br>Includes intent to:<br>• Impair the operation of any PC<br>• Prevent or hinder access to a program<br>• Impair the operation of any program or reliability of data. |

## L.O.R. (*) Questions (cont)

| | |
|---|---|
| Copyright, Designs & Patents Act | Protects individuals/organisations intellectual data. It protects:<br>• Income for the authors - Allows the author to license the data.<br>• Cost of creating the product - some products can cost thousands to produce<br>• Quality of Produce - pirates often alter products to bypass security<br>• Alteration Protection - Altering programs can have unintended aftereffects. |

## L.O.R. (*) Questions (cont)

| | |
|---|---|
| Regulation of Investigatory Powers Act | This act allows government agencies, to request access to secure information. It makes provisions for:<br>• Interception of communication<br>• Acquisition and disclosure of data<br>• Surveillance<br>• Access to electronic data protected by encryption. |

## L.O.R. (*) Questions (cont)

## L.O.R. (*) Questions (cont)

**Moral/Ethical/Social Issues**

## L.O.R. (*) Questions (cont)

| | |
|---|---|
| Computers in the Workplace | • Big Brother concern - an employer could watch over employees<br>• Reduced Productivity - employees can do multiple things at once, which may reduce productivity as employees may 'waste' time |

By **0llieC** (0llieC)
cheatography.com/0lliec/

Published 20th February, 2019.
Last updated 12th June, 2017.

## L.O.R. (*) Questions (cont)

| | |
|---|---|
| Automated Decision Making | Computers are starting to have the ability to make decisions based on input data. Usually it can be better than any human making the same decision. The issue is what happens when the wrong decision is made, who is to blame? |

## L.O.R. (*) Questions (cont)

| | |
|---|---|
| AI | This is the one of the biggest issues, as AI use is rising among recent years. The issues are the same as automated decision making, but more issues arise when you consider cognitive/when is a computer considered alive? |

## L.O.R. (*) Questions (cont)

| | |
|---|---|
| Environment | The increase in use of computers = more RAW materials Another issue is the disposal of old parts/-devices. |

By **0llieC** (0llieC)
cheatography.com/0lliec/

Published 20th February, 2019.
Last updated 12th June, 2017.

## L.O.R. (*) Questions (cont)

| | |
|---|---|
| Censorship | Moral concerns are raised at whether the internet should be censored, would it be restricting the freedom of information. The issues arise when considering adult content, and piracy. |

## L.O.R. (*) Questions (cont)

| | |
|---|---|
| Monitoring Behaviour | It is possible to monitor what individuals are using a computer for, there is a moral issue when considering how much should an individual be monitored, and the issues based on a persons privacy. |

## L.O.R. (*) Questions (cont)

| | |
|---|---|
| Personal Inform-ation | Rises a privacy concern. Computers can now monitor peoples information and collect it. When does this become a breach of privacy. |

By **0llieC** (0llieC)
cheatography.com/0lliec/

Published 20th February, 2019.
Last updated 12th June, 2017.

## L.O.R. (*) Questions (cont)

| | |
|---|---|
| Piracy | Breaking the law (C.D.P.) but people do it anyway. |

## L.O.R. (*) Questions (cont)

| | |
|---|---|
| Offensive Material | Computers are general purpose, what people do with them can be considered offensive/morally wrong e.g. cyberbullying, which can have drastic effects |

## Data Structures

| | |
|---|---|
| Array | A data structure used to whole elements of the same data type. |

By **0llieC** (0llieC)
cheatography.com/0lliec/

Published 20th February, 2019.
Last updated 12th June, 2017.

## Data Structures (cont)

| | |
|---|---|
| 1D Array | An Array with a single dimension, i.e. it only has a given length |

## Data Structures (cont)

| | |
|---|---|
| 2D Array | An array with 2 dimensions. It is commonly used to represent coordinates or a table, with the indexes relating to rows/columns |

## Data Structures (cont)

| | |
|---|---|
| 3D Array | An array with 3 dimensions. It is used for representing 3D space so is also used for coordinates a lot. |

By **0llieC** (0llieC)
cheatography.com/0lliec/

Published 20th February, 2019.
Last updated 12th June, 2017.

## Data Structures (cont)

| | |
|---|---|
| Linked List | A data structure where each element in the list points to the next one. This makes is very easy to add/remove/reorder elements as only the pointer needs to change each time. |

## Data Structures (cont)

| | |
|---|---|
| Queue | A First In, First Out (FIFO) data structure.<br>When coding a queue, there must be the possibility to:<br>• Check if the queue is full<br>• Read/Remove/Return an element from the front of the Q<br>• Place a new element at the end of the Q |

## Data Structures (cont)

| | |
|---|---|
| Circular Queue | The end of a queue linking back to the beginning. |

By **0llieC** (0llieC)
cheatography.com/0lliec/

Published 20th February, 2019.
Last updated 12th June, 2017.

## Data Structures (cont)

| | |
|---|---|
| Stack | A First In, Last Out (FILO) data structure.<br>When coding a stack, there must be the possibility to:<br>• Check if the stack is full/empty<br>• Read/Remove/Return an element from the top of the stack (pop)<br>• Add a new value to the top of the stack (push) |

## Data Structures (cont)

| | |
|---|---|
| Graph | A set of nodes/vertices connected by edges. |

## Data Structures (cont)

| | |
|---|---|
| Direction-al-Graph | A graph where the edges have a direction. |

By **0llieC** (0llieC)
cheatography.com/0lliec/

Published 20th February, 2019.
Last updated 12th June, 2017.

## Data Structures (cont)

| | |
|---|---|
| Bi-Direct-ional Graph | A graph where the edges have 2 way directions. ↔ |

## Data Structures (cont)

| | |
|---|---|
| Trees | A tree is a simple un-directed graph which contains no loops. A tree has a root where all other nodes/edges originate from |

## Data Structures (cont)

| | |
|---|---|
| Binary Tree | A tree where each node has a maximum of 2 sub-nodes. Nodes with no child nodes are called leafs, and the edges, branches. |

By **0llieC** (0llieC)
cheatography.com/0lliec/

Published 20th February, 2019.
Last updated 12th June, 2017.

## Data Structures (cont)

| | |
|---|---|
| Hash Table | A table where the index system is the data the person is looking for, but |

## Input/Output/Storage

| | |
|---|---|
| Input Device | A device (piece of computer hardware equipment) that is used to provide data and control signals to an information processing system such as a computer or information appliance. **Examples of input devices include keyboards, mouse, scanners, digital cameras and joysticks.** |

## Input/Output/Storage (cont)

| | |
|---|---|
| Output Device | is any device used to send data from a computer to another device or user. Most computer data output that is meant for humans is in the form of audio or video. **Examples include monitors, projectors, speakers, headphones and printers.** |

By **0llieC** (0llieC)
cheatography.com/0lliec/

Published 20th February, 2019.
Last updated 12th June, 2017.

## Input/Output/Storage (cont)



By **0llieC** (0llieC)
cheatography.com/0lliec/

## Input/Output/Storage (cont)

### Memory

Published 20th February, 2019.
Last updated 12th June, 2017.


## Input/Output/Storage (cont)

## Input/Output/Storage (cont)

| | |
|---|---|
| HDD/Magnetic | Information is held in blocks consisting of tracks and sectors. Each block contains the same amount of information, therefore information is more dense closer to the centre.<br><br>**Rotation Speed**<br>• A HDD consists of a very fast spinning disk (5400 - 7200 rpm)<br>• A reading head is suspended above the disk due to the Bernoulli effect<br>• Due to fast speeds, the housing **has** to be evacuated<br><br>**Capacity vs Cost**<br>• Largest HDD avaliable ~ 12TB.<br>• Roughly 3p per GB. (£0.00000-0000027915 per Byte) |

## Input/Output/Storage (cont)

| | |
|---|---|
| SDD/Flash | A storage medium that has no moving parts. It uses a data controller to control the read/write of data. 2 rules of the data controller:<br>**1.** You can combine pages to form a block, but a block **cannot** overwrite individual pages<br>**2.** Before writing to a memory location, the page previously allocated must be erased.<br><br>**Pros**<br>• Low Latency Time<br>• Fast Transfer speed<br><br>**Cons**<br>• More Expensive<br><br>**Capacity vs Cost**<br>• Largest: ~4TB<br>• Roughly 30p per GB (£0.00-0000000291625 per Byte) |

## Input/Output/Storage (cont)

| | |
|---|---|
| Disc/Optical | A storage medium that uses binary pits to encode data. A laser is beamed at the disk and uses the diffraction of the light to detect a 0/1 (trough/peak).<br><br>**Read-only**: A laser is used to burn the disks, the data cannot be changed<br>**Re-Writable**: A dye is used where if a high temp is used, it will go opaque (creating a peak - 1) and if a higher temp is used it goes transparent (a trough - 0). The disk is now reusable. |

## Input/Output/Storage (cont)

Speeds

**Solid State**: 200 to 2500 MB/s
**Hard-Drive**: 1030 MB/s
**Optical (x1 Speeds)**:
- Blue-ray: 4.29 MB/s
- DVD: 1.32 MB/s
- CD: 0.15 MB/s

## Input/Output/Storage (cont)

## Input/Output/Storage (cont)

RAM

The 'working' area of the computer. Programs and data currently in use is stored in the RAM. On startup the BIOS loads the OS into the RAM.

**Characteristics**

• Random Access - allows data items to be read or written in almost the same amount of time irrespective of the physical location of data inside the memory.
• **Volatile** → emptied on power down
• ~1-16GB

By **0llieC** (0llieC)
cheatography.com/0lliec/

Published 20th February, 2019.
Last updated 12th June, 2017.

## Input/Output/Storage (cont)

| | |
|---|---|
| ROM | A permanent area of storage. The contents cannot be altered by software. Contents of ROM is written at manufacture<br><br>**Characteristics**<br>• Read-Only Access<br>• **Non-Volatile** → retains data at power down<br>• Mainly used to store firmware or application software in plug-in cartridges.<br>• ~4MB<br>• Examples of ROM: Bootloader (BIOS/UEFI), |

## Input/Output/Storage (cont)

| | |
|---|---|
| BIOS/UEFI | Basic Input Output System/Unified Extensible Firmware Interface The BIOS is preforms the hardware initialisation during the bootup, and provides runtime services between the OS and hardware. UEFI was designed to be the successor to the BIOS |

## Input/Output/Storage (cont)

| | |
|---|---|
| Virtual Memory | When the RAM is full, the OS uses some of the secondary storage as Virtual Memory. This means the computer can continue to run. Pages (blocks of data) are transferred to the virtual memory when not needed thus freeing up space, and returned to RAM when they are needed. |

By **0llieC** (0llieC)
cheatography.com/0lliec/

Published 20th February, 2019.
Last updated 12th June, 2017.

## Operating System

| | |
|---|---|
| Operating System | Software that provides:<br>• Process Management<br>• Memory Management<br>• Device Management<br>• User Interface<br>• File Management<br><br>**Fundamentally its software that manages/interfaces computer hardware and software** |

## Operating System (cont)

| | |
|---|---|
| Kernel | The very core of the OS that provides the interface between the user and the hardware. Applications use the kernel to send/receive data from hardware. |

## Operating System (cont)

| | |
|---|---|
| Memory Management | A OS must manage the computers memory including adding/removing programs and data from RAM, allowing multiple programs to be run at the same time. The OS also reallocates memory when it is no-longer in use (i.e. when a program is closed)<br><br>**Paging vs Segmentation & Virtual Memory**<br>• Segmentation. → Memory is split into variable sized blocks, and programs are segmented, with each segment being a logical divider. A segment table then maps segments onto memory blocks. Generally slower than paging due to the placement algorithm<br>• Paging. → RAM is split into fixed sized blocks - frames. Programs are split into same-sized blocks - pages. Any page can be placed in any frame, easy to allocate as all equal size.<br>• If the RAM is full. Pages are transferred to the secondary storage acting as memory - Virtual Memory. Pages are moved in/out as needed.<br>• Thrashing → is when pages are being constantly swapped between RAM and V.Mem. It can cause speed issues as the secondary storage's speed << RAM's speed. |

## Operating System (cont)

| | |
|---|---|
| Interrupts | Interrupts are a form of error checking. If an error occurs, the interrupt is stored in a priority queue. After the next instruction has been executed, the interrupt queue is checked for any interrupt and the processor runs a set of instructions called the Interrupt Service Routine (ISR), with each interrupt having its own ISR. Before the ISR is run, the current values in the registers are stored, so that the processor can return to its previous position. Examples of interrupt types are:<br>• I/O Interrupt → A status of a channel has changed, Occurs when an IO operation is complete or a device is ready.<br>• Timer Interrupt → Allows the processor to preform tasks at intervals<br>• Program Check → Most commonly memory access violations - accessing memory that doesn't exist or is not in use<br>• Machine Check → when hardware |

## Operating System (cont)

| | |
|---|---|
| Process Management | Involves the scheduling and switching of programs and threads. Modern PCs have 'multitasking' but it is just clever scheduling. |

## Operating System (cont)

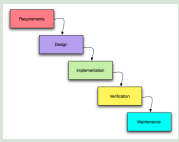
| | |
|---|---|
| Scheduling Techniques | **First Come, First Served**<br>As the name suggests.<br>• Poor Efficiency<br><br>**Round Robin**<br>Each process has a set number of processing time. Processor switches in a circular fashion<br>• Easy Implementation<br>• Can be inefficient<br>• Time can be lost waiting for inputs<br><br>**Shortest Job First**<br>The process with the shortest processing time is processed<br>• Long Process can be waiting a long time - processor starvation<br><br>**Shortest time *remaining***<br>The process with the shortest remaining processing time is processed. If another job with a shorter time remaining arrives, it will switch<br>• Short jobs executed quickly<br>• Starvation can still occur<br><br>**Multi-Level Queue**<br>Processes are given a priority when they arrive, dep. on their time remaining, process type and memory size.<br>• Important jobs processed first<br><br>**Multi-Level Feedback Queue**<br>Same as a MLQ but the processor can change the priority of a process, most likely due to a process taking up too much processing time.<br>• Stops starvation<br>• Allows interactivity<br>• Priorities can be changed |

## Operating System (cont)

## Operating System (cont)

| | |
|---|---|
| Device Management | The OS can make devices acessible to other programs through the use of Device Drivers. It is a piece of software that controls the hardware and provides the interface so that programs and the OS can use the device. Devices cause interrupts on the processor and depending on its priority is when the interrupt is processed. |

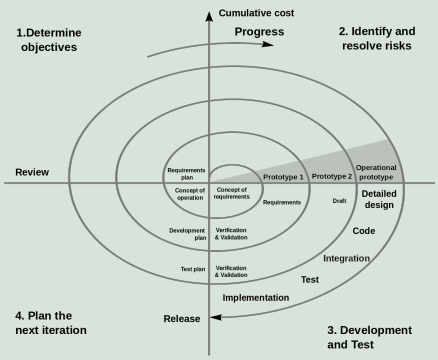
## Waterfall Method



Each of the stages are classified as *milestones*. Following the methodology strictly would mean the system is developed flowing down the waterfall. Another version exists where there is iteration back up the steps.

| | |
|---|---|
| Types of OS | **Embedded**<br>• Mostly hidden in devices, generally within the hardware themselves.<br>• Built into objects<br>• Have a dedicated purpose<br>• Little/no user interface<br>• Fully Autonomous<br>• Use limited resources - only whats required<br><br>**Multi-Tasking**<br>• Several programs/processes at the same time (concurrent).<br>• Can either be process management or through parallel processing<br>• Most General Purpose OS' are now Multitasking<br><br>**Multi-User**<br>• Must be a multi-tasking OS too<br>• Several users accessing the processor/programs/resources at the same time.<br>• Usually a round robin approach.<br>• Shared processing.<br><br>**Real-Time**<br>• Inputs being processed under strict time limits. For requirements:<br>**1.** Support Non-Sequential programs<br>**2.** Handle parallel and unpredictable events<br>**3.** Produce responses within the time limit<br>**4.** Have fail-safes to guarantee response time<br><br>**Distributed**<br>A collection of independent nodes, each with its own hardware. The OS presents the systems as an individual. For example: AI; Weather Forecasting; Online Shopping. Each may have the main system on one server, and other things processed on another. The pros of this are that it reduces the load on one computer, and if one fails, it may be able to continue. |

## Spiral Method

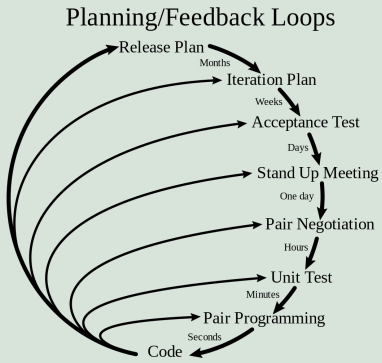



The method starts in the centre and spirals outwards. The purpose is to elimin-ate/reduce any project failures by constantly returning to each of the milestones. The review stage is where the client is consulted with to determine the progress.

## RAD


This development methodology requires minimal documentation, but requires a high amount of involvement of the client as a prototype is created, then reviewed then improved upon.

## Extreme Programming

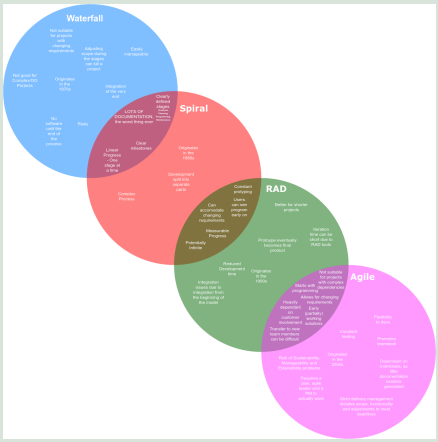



This is one on the agile approaches to software development. It allows for client changes throughout the life cycle and the constant review of progress and client involvement give it its name as 'extreme'.

By **0llieC** (0llieC)
cheatography.com/0lliec/

Published 20th February, 2019.
Last updated 12th June, 2017.

## Methodology Comparison




## Programming paradigms

| | |
|---|---|
| Object Orientated | Code is divided into objects which possess state and behaviour. Follows the principles of encapsulation, abstraction, inheritance and polymorphism. |

## Programming paradigms (cont)

| | |
|---|---|
| Logic | The code consists of a series of rules which define a scenario. Answers can be obtained by asking questions in a specific format. |

By **0llieC** (0llieC)
cheatography.com/0lliec/

Published 20th February, 2019.
Last updated 12th June, 2017.

## Programming paradigms (cont)

| | |
|---|---|
| Data Query Languages | Queries to a database or other data structure are specified by what is wanted rather than how to get it. |

## Programming paradigms (cont)

| | |
|---|---|
| Scripting | Code is written to automate processes rather than create entire applications. Scripting languages are often embedded into other systems. |

## Programming paradigms (cont)

| | |
|---|---|
| Procedural | Allows structured programming with sequence, selection, iteration and recursion. Code can be made modular with the use of procedures. |

By **0llieC** (0llieC)
cheatography.com/0lliec/

Published 20th February, 2019.
Last updated 12th June, 2017.

## Programming paradigms (cont)

| | |
|---|---|
| Functional | Code is divided into isolated functions. There is no global state, only arguments and return values are important. Closely linked to mathematics. |

## Programming paradigms (cont)

| | |
|---|---|
| Assembly Languages | One to one correspondence between lines of code and processor instructions. Unlike raw machine code however, you can have variable names and labels. |

## Compression

| | |
|---|---|
| Lossless | Compressing a file without the loss of data |

By **0llieC** (0llieC)
cheatography.com/0lliec/

Published 20th February, 2019.
Last updated 12th June, 2017.

## Compression (cont)

| | |
|---|---|
| Lossy | Compressing a file by removing redundant data. |

## Compression (cont)

| | |
|---|---|
| Run Length Encoding (RLE) | RLE identifies repeating patterns of data and stores a copy of the information and how many times it occurs in succession. |

## Compression (cont)

| | |
|---|---|
| Dictionary-based | Uses a substring search to match strings in the file to be compressed to those stored in a dictionary. If a match is found then the string is substituted for the dictionary index. If no match is found, the string is added to the dictionary |

By **0llieC** (0llieC)
cheatography.com/0lliec/

Published 20th February, 2019.
Last updated 12th June, 2017.

## Encryption and Hashing

| | |
|---|---|
| Symmetric Encryption | Encryption that uses the same key to both encrypt and decrypt.<br>**Uses**: Encrypted Harddrives |

## Encryption and Hashing (cont)

| | |
|---|---|
| Asymmetric Encryption | Encryption where different keys encrypt and decrypt the data.<br>**Uses**: Online transactions |

## Encryption and Hashing (cont)

| | |
|---|---|
| Client-Server Commun-ication | The client generates a **session key** and uses the **public key** to encrypt it<br>↓<br>The server decrypts the session key with the **private key**<br>↓<br>Client-Server now commun-icate using **symmetric encryption with the session key** |

By **0llieC** (0llieC)
cheatography.com/0lliec/

Published 20th February, 2019.
Last updated 12th June, 2017.

## Encryption and Hashing (cont)

| | |
|---|---|
| Private Key | The private key consists of 2 **very** large prime numbers |

## Encryption and Hashing (cont)

| | |
|---|---|
| Public Key | The public key is the product of the 2 prime numbers making the private key. As no efficient non-quantum integer factorisation algorithm exists, it is practically impossible to crack the private key by brute force. |

## Encryption and Hashing (cont)

| | |
|---|---|
| Hashing | Using an algorithm to map data of any size to a fixed size. Unlike encryption, hashing **cannot** be undone, it is therefore a lossy process.<br>**Uses**:<br>• Rapid data access in a hash table<br>• Error checking and corruption detection - such as downloads<br>• Password verification - the plain-text password would not have to be stored<br><br>*A good hash algorithm:**<br>• Same message = Same hash<br>• Quick to compute<br>• Impossible to generate a message from the hash<br>• A small change = a big hash change<br>• Impossible to find 2 messages with the same hash |

## Databases

| | |
|---|---|
| Database | A structured system to hold data |

## Databases (cont)

| | |
|---|---|
| Relational Database | a database structured to recognize relations between stored items of information. |

## Databases (cont)

| | |
|---|---|
| Flat File Database | A flat file database is a database that stores data in a plain text file. Each line of the text file holds one record, with fields separated by delimiters, such as commas or tabs. While it uses a simple structure, a flat file database cannot contain multiple tables like a relational database can. |

By **0llieC** (0llieC)
cheatography.com/0lliec/

Published 20th February, 2019.
Last updated 12th June, 2017.

## Databases (cont)

| | |
|---|---|
| Entity | Any item about which data is stored e.g. Student, Pizza, Stock etc. |

## Databases (cont)

| | |
|---|---|
| Attribute | A feature of the entity |

## Databases (cont)

| | |
|---|---|
| Foreign Key | A unique identifier to each record held in the relational database. |

By **0llieC** (0llieC)
cheatography.com/0lliec/

Published 20th February, 2019.
Last updated 12th June, 2017.

## Databases (cont)

| | |
|---|---|
| Composite Primary Key | A combination of 2+ fields that act as a primary key |

## Databases (cont)

| | |
|---|---|
| Foreign Key | A way to build a relationship between 2 tables, the foreign key is another tables primary key |

## Databases (cont)

| | |
|---|---|
| Secondary Key | A key that is indexed to allow for faster searching. There can be multiple secondary keys and they don't have to be unique. |

By **0llieC** (0llieC)
cheatography.com/0lliec/

Published 20th February, 2019.
Last updated 12th June, 2017.

## Databases (cont)

Inner Join — Combining columns from one+ tables by using values common to each.

```
SELECT table1.column1,
table2.column2...
FROM table1
INNER JOIN table2
ON table1.common_field =
table2.common_field;
```

## Databases (cont)

## Databases (cont)

**Normal Forms**

By **0llieC** (0llieC)
cheatography.com/0lliec/

Published 20th February, 2019.
Last updated 12th June, 2017.

## Databases (cont)

## Databases (cont)

1NF
- Each row is unique - it has a primary key
- Each column has a unique name
- No columns with similar or repeated data (i.e. choice1, choice2 etc.)
- Each data item cannot be broken up any further - no commas in the data

## Databases (cont)

2NF
- 1NF
- If the primary key is a composite of attributes (contains multiple columns), the non-key attributes (columns) must depend on the whole key.

By **0llieC** (0llieC)
cheatography.com/0lliec/

Published 20th February, 2019.
Last updated 12th June, 2017.

## Databases (cont)

3NF
- 1NF
- 2NF
- There are no non-key attributes that depend on other non-key attributes

## Databases (cont)

## Databases (cont)

**CRUD**

By **0llieC** (0llieC)
cheatography.com/0lliec/

Published 20th February, 2019.
Last updated 12th June, 2017.

## Databases (cont)

≡

## Databases (cont)

| | |
|---|---|
| CREATE | INSERT INTO tableName (fieldNames) VALUES (values) |

## Databases (cont)

| | |
|---|---|
| READ | SELECT fieldNames FROM tableName WHERE fieldName = value ORDER BY fieldName |

By **0llieC** (0llieC)
cheatography.com/0lliec/

Published 20th February, 2019.
Last updated 12th June, 2017.

## Databases (cont)

| | |
|---|---|
| UPDATE | UPDATE tableName SET fieldName = value WHERE fieldName = value |

## Databases (cont)

| | |
|---|---|
| DESTROY | DELETE FROM tableName where fieldName = value<br><br>DROP tableName |

## Databases (cont)

By **0llieC** (0llieC)
cheatography.com/0lliec/

Published 20th February, 2019.
Last updated 12th June, 2017.

## Databases (cont)

**ACID Principles (Transactions)**

## Databases (cont)

## Databases (cont)

| | |
|---|---|
| Atomicity | Transactions are either done, or not done. **Never partially applied** |

By **0llieC** (0llieC)
cheatography.com/0lliec/

Published 20th February, 2019.
Last updated 12th June, 2017.

## Databases (cont)

| | |
|---|---|
| Consistency | Referential Integrity and other constraints must be adhered to |

## Databases (cont)

| | |
|---|---|
| Isolation | Transactions preformed simultaneously must have the same result as if they were preformed sequentially |

## Databases (cont)

| | |
|---|---|
| Durability | Transactions that have been committed must be done fully and remain so. |

By **0llieC** (0llieC)
cheatography.com/0lliec/

Published 20th February, 2019.
Last updated 12th June, 2017.

## Databases (cont)

By **0llieC** (0llieC)
cheatography.com/0lliec/

## Databases (cont)

### Concurrent Accessing

Published 20th February, 2019.
Last updated 12th June, 2017.


## Databases (cont)

## Databases (cont)

| | |
|---|---|
| Concurrent Access | Is ensuring that more than one user can at least view data at the same time. |

## Databases (cont)

| | |
|---|---|
| Record Locking | Making a file read-only to anybody else who opens the file while changes are being made. |

## Databases (cont)

| | |
|---|---|
| Deadlock | When 2 separate transactions lock the file the other transaction needs, thus both are in a state of waiting. |

By **0llieC** (0llieC)
cheatography.com/0lliec/

Published 20th February, 2019.
Last updated 12th June, 2017.

## Databases (cont)

| | |
|---|---|
| Serial-isation | Create a clone of the data item, so the user can make changes, then upload a copy of the clone to the database. This will ensure that no updates or changes can be lost due to uploading a copy of the local version. |

## Databases (cont)

| | |
|---|---|
| Timestamp Ordering | A non-lock way of concurrent access, so multiple people can access the data at one time. The main process is that the lower timestamps occur first. |

## Networks

| | |
|---|---|
| Standard | A definition or a format that has been approved by a recognised standards organisation. **de jeur** (by force of law) or **de facto** a standard that has just been accepted over time |

By **0llieC** (0llieC)
cheatography.com/0lliec/

Published 20th February, 2019.
Last updated 12th June, 2017.

## Networks (cont)

**Protocol** An agreed-upon format for exchanging data between devices. It determines:

- The error checking used
- Compression method, if any
- How the sender will indicate end of transmission
- How the receiver will indicate the data has been received.

## Networks (cont)

**LAN** **Local Area Network**

- Geographically Small (buildings/a site)
- Equipment is generally owned by the company/people using it
- Generally Faster
- Uses layers 1 and 2 devices - hubs/switches

## Networks (cont)

**WAN** **Wide Area Network**

- Geographically remote (across a country/between continents/the w.w.w.)
- Connects LANs together with third party telecommunication equipment
- Slower speed than LAN
- Uses layer 3 devices - routers/multi-layer switches

By **0llieC** (0llieC)
cheatography.com/0lliec/

Published 20th February, 2019.
Last updated 12th June, 2017.

## Networks (cont)

**Network Topologies**

**Bus Topology**
• All devices connected to a central cable (backbone)
• Devices have equal rights
• Collisions can occur if multiple devices send data at once

**Star Topology**
• A hub at the centre of the network. Requests are sent to all other devices connected to is
• The hub reads the packets and determines the MAC address of the recipient

**Ring Topology**
• A token is passed around the ring until one of the devices requests to use it.
• The token is filled with the frame of data
• It is passed around the network to each device until it reaches the recipient
• Recipient acknowledges the data has arrived.

## Networks (cont)

**Client-Server**

• One entity (client) requests services from another (server)
• Server stores security information e.g. logins and permissions.

**Pros**
+ Centralised control
+ Single data storage
+ Easy backing up and restoring
+ Remote access
+ Can define security rights and permissions

**Cons**
- Too many requests can cause congestion
- If the server fails, whole network goes down
- Expensive to install and manage
- Requires professionals to install and manage

## Networks (cont)

**Peer-to-Peer**

• All computers have equal rights and act as both a client and server
• Popular applications include the BitTorrent Network, and BitCoin

**Pros**
+ Easy to set up
+ More reliable as central dependencies are eliminated
+ No-need for a system administrator as every user is the admin of their machine
+ Cheaper to implement and maintain.

**Cons**
- Difficult to administor as there is no central dependency
- Less security therefore viruses and other malware can easily be transmitted
- Data recovery is difficult as there is no central storage, each computer requires ots own backup system
- "(Lots of movies, tv shows and music are transferred using P2P, via torrents)"

By **0llieC** (0llieC)
cheatography.com/0lliec/

Published 20th February, 2019.
Last updated 12th June, 2017.

## Networks (cont)

| | |
|---|---|
| Packet Switching | A message/data is broken into a number of parts (packets) which are sent independently, over whatever route is optimum for each packet, and reassembled at the destination.<br><br>**Pros**<br>+ Efficient use of a network<br>+ Can easily circumvent broken sections of a network<br>+ Network only has to increase slowly as demand does<br><br>**Cons**<br>- Time taken to rebuild packets is variable - an issue for time-sensitive data<br>- Not good for small data. |

## Networks (cont)

| | |
|---|---|
| Circuit Switching | Communication where a dedicated channel (or circuit) is established for the duration of a transmission.<br><br>**Pros**<br>+ Data arrives in order sent<br>+ No additional information has to be added - e.g. headers<br><br>**Cons**<br>- Portion of the network is unavailable while in use<br>- Data is easily intercepted. |

## Networks (cont)

| | |
|---|---|
| Domain Name System (DNS) | A system that converts the web address: www.website.something into the IP address of the host server. |

By **0llieC** (0llieC)
cheatography.com/0lliec/

Published 20th February, 2019.
Last updated 12th June, 2017.

## Networks (cont)

| | |
|---|---|
| MAC Address | Unique 6-byte identifier that is given to NICs. Assigned to the NIC by the manufacturer. |

## Networks (cont)

| | |
|---|---|
| IPv4 | Most commonly used IP version. Its a 32-bit system, so there are $2^{32}$ addresses available. |

## Networks (cont)

| | |
|---|---|
| IPv6 | IPv6 is a 128-bit address, so there are $2^{128}$ addresses available. |

By **0llieC** (0llieC)
cheatography.com/0lliec/

Published 20th February, 2019.
Last updated 12th June, 2017.

## Networks (cont)



By **0llieC** (0llieC)
cheatography.com/0lliec/

## Networks (cont)

### TCP/IP Stack

Published 20th February, 2019.
Last updated 12th June, 2017.


## Networks (cont)

## Networks (cont)

**Application Layer**

↓

## Networks (cont)

The Application layer ensures the data is sent in an understandable format by the recipient. It formats the data to meet the standards of the protocol.

## Networks (cont)

**Transport Layer**

↓

By **0llieC** (0llieC)
cheatography.com/0lliec/

Published 20th February, 2019.
Last updated 12th June, 2017.

## Networks (cont)

The transport layer takes the data and splits it into data packets. Each one is given a number, specifying the order so it can be reconstructed. The port number is also added depending on the application being used for example HTTP is port 80.

## Networks (cont)

**Network Layer**



## Networks (cont)

The network layer is where the IP of the sender is attached, so the recipient can send a message saying the packets were received. It also attaches the recipients IP. This is also the layer where the Time To Live (TTL) is added to the header. It governs how many times the packet can hop before deleting itself, this ensure infinite loops don't occur.

By **0llieC** (0llieC)
cheatography.com/0lliec/

Published 20th February, 2019.
Last updated 12th June, 2017.

## Networks (cont)

**Link Layer**

↓

## Networks (cont)

This is the layer where the MAC address of both the sender and recipient is attached, allowing the packets to be directed to a specific NIC.

## Networks (cont)

☰

By **0llieC** (0llieC)
cheatography.com/0lliec/

Published 20th February, 2019.
Last updated 12th June, 2017.

## Networks (cont)

**Internet/Network Protocols**

## Networks (cont)

## Networks (cont)

| | |
|---|---|
| HTTP (Hypertext Transfer Protocol) | Defines how webpages are transferred from server to the client. The HTTP will make a request to the IP and the server responds with a webpage. There are 8 different HTTP commands including **GET**, **POST** and *CONNECT** |

By **0llieC** (0llieC)
cheatography.com/0lliec/

Published 20th February, 2019.
Last updated 12th June, 2017.

## Networks (cont)

| | |
|---|---|
| HTTPS (Secure Hypertext Transfer Protocol) | Connects via a different port and encrypts the data between the HTTP and the TCP protocols. |

## Networks (cont)

| | |
|---|---|
| FTP (File Transfer Protocol) | The protocol used to download and upload files. Most modern browsers have built in FTP |

## Networks (cont)

| | |
|---|---|
| POP3 (Post Office Protocol 3) | Allows emails to be received from a server. The protocol connects to the email server, downloads a local copy then deletes them from the server. |

By **0llieC** (0llieC)
cheatography.com/0lliec/

Published 20th February, 2019.
Last updated 12th June, 2017.

## Networks (cont)

| | |
|---|---|
| SMTP (Simple Mail Transfer Protocol) | Used for sending emails |

## Networks (cont)

| | |
|---|---|
| SSH (Secure Shell) | Remote-access protocol, allows secure communication between a client and server |

## Networks (cont)

| | |
|---|---|
| CSMA/CA (Carrier Sense Multiple Access with Collision Avoidance) | Is a transmission protocol that prevents packet collisions. Once it receives a packet, it checks whether the channel is clear, if it is not available it will generate a random wait time, when it will check again.<br><br>If a packet is larger than the permitted size is needed to be sent, a *handshake* needs to occur first - the RTS/CTS (Request to Send/Clear to Send) protocol. This protocol only occurs when the packet is larger than the threshold. |

By **0llieC** (0llieC)
cheatography.com/0lliec/

Published 20th February, 2019.
Last updated 12th June, 2017.

By **0llieC** (0llieC)
cheatography.com/0lliec/



## Network Security

Published 20th February, 2019.
Last updated 12th June, 2017.

## Networks (cont)

| | |
|---|---|
| Virus | A embedded program intended to cause damage to a PC. It copies itself onto the disk and hides itself. It attempts to duplicate itself and spread to other computers. |

## Networks (cont)

| | |
|---|---|
| Worm | A virus but it is contained within its own program. |

## Networks (cont)

| | |
|---|---|
| Trojan | A non-self-replicating virus hidden in a downloaded file, and unleashed on execution. |

By **0llieC** (0llieC)
cheatography.com/0lliec/

Published 20th February, 2019.
Last updated 12th June, 2017.

## Networks (cont)

| | |
|---|---|
| Ransom Ware | A trojan/worm that encrypts data and then charges the owner to decrypt it. |

## Networks (cont)

| | |
|---|---|
| Firewall | The purpose of a firewall is to control the traffic flowing in and out of a network. It can be hardware or software based, and sometimes is a combination of both. It can be setup to block individual website addresses or specific computers. |

## Networks (cont)

| | |
|---|---|
| Proxy Server | f a user requests a service from the network, it is first passed to the proxy, before the proxy server then performs the request on the behalf of the network user. If the resource is banned the request can be rejected, There is never any direct contact between user and resource, as the proxy acts as a "middle man". |

By **0llieC** (0llieC)
cheatography.com/0lliec/

Published 20th February, 2019.
Last updated 12th June, 2017.

## Networks (cont)

| | |
|---|---|
| WPA/WPA2 | It requires you to enter a password when accessing a network. It acts as layer of protection. |

## Networks (cont)

## Networks (cont)

### Network Hardware

By **0llieC** (0llieC)
cheatography.com/0lliec/

Published 20th February, 2019.
Last updated 12th June, 2017.

## Networks (cont)

## Networks (cont)

| | |
|---|---|
| Hub | It receives a signal from a node and transmits it to all the other nodes. It is cheap and effective for small networks but for larger networks causes too many collisions |

## Networks (cont)

| | |
|---|---|
| Switch | A switch has a small amount of internal memory, that allows it to generate a look-up table. When data is sent the switch finds the appropriate node. Unlike a hub, it **doesn't** send the data to all the nodes, just the receiver. |

By **0llieC** (0llieC)
cheatography.com/0lliec/

Published 20th February, 2019.
Last updated 12th June, 2017.

## Networks (cont)

| | |
|---|---|
| Router | A device that forwards packets from one network to another. |

## Networks (cont)

| | |
|---|---|
| Gateway | The entrance and exit of networks. The main use is to connect multiple networks with different architectures |

## Search Engines

| | |
|---|---|
| Overall Search engine | Crawling the web with 'spiders' TF-IDF (Term Frequency - Inverse Document Frequency) The PageRank algorithm Other factors, such as domain name, page age, mobile friendliness |

By **0llieC** (0llieC)
cheatography.com/0lliec/

Published 20th February, 2019.
Last updated 12th June, 2017.

## Search Engines (cont)

## Search Engines (cont)

| | |
|---|---|
| Page Rank | PageRank works by counting the number and quality of links to a page to determine a rough estimate of how important the website is. The underlying assumption is that more important websites are likely to receive more links from other websites. |

## Data Types

| | |
|---|---|
| Integer | A whole number |

By **0llieC** (0llieC)
cheatography.com/0lliec/

Published 20th February, 2019.
Last updated 12th June, 2017.

## Data Types (cont)

| | |
|---|---|
| Real/Float | Float is a term is used in various programming languages to define a variable with a fractional value. Numbers created using a float variable declaration will have digits on both sides of a decimal point. This is in contrast to the integer data type, which houses an integer or whole number. |

## Data Types (cont)

| | |
|---|---|
| Boolean | A value with a True or False condition - can possibly use 0/1 instead |

## Data Types (cont)

| | |
|---|---|
| Character | A single keyboard/unicode character |

By **0llieC** (0llieC)
cheatography.com/0lliec/

Published 20th February, 2019.
Last updated 12th June, 2017.

## Data Types (cont)

| | |
|---|---|
| String | A set of characters, used to store text. |

## Data Types (cont)

| | |
|---|---|
| Date/Time | A representation of time. Can be represented in either text or number format |

## Data Types (cont)

By **0llieC** (0llieC)
cheatography.com/0lliec/

Published 20th February, 2019.
Last updated 12th June, 2017.

## Data Types (cont)

**Number Bases**

## Data Types (cont)

| | |
|---|---|
| Binary (Base 2) | Groups of bits in 1 and 0 (2 possible values). It uses positional numbering but with powers of 2, not 10 (denary numbers - normal) |

## Data Types (cont)

| | |
|---|---|
| Hexadecimal (Base 16) | 0 → 0000 → 0<br>1 → 0001 → 1<br>2 → 0010 → 2<br>3 → 0011 → 3<br>4 → 0100 → 4<br>5 → 0101 → 5<br>6 → 0110 → 6<br>7 → 0111 → 7<br>8 → 1000 → 8<br>9 → 1001 → 9<br>10 → 1010 → A<br>11 → 1011 → B<br>12 → 1100 → C<br>13 → 1101 → D<br>14 → 1110 → E<br>15 → 1111 → F |

By **0llieC** (0llieC)
cheatography.com/0lliec/

Published 20th February, 2019.
Last updated 12th June, 2017.

## Data Types (cont)

| | |
|---|---|
| Denary (Base 10) | Normal number formats with the positional numbering in powers of 10. |

## Data Types (cont)

| | |
|---|---|
| Sign & Magnitude | A form of showing negative binary numbers where the first bit is the sign (0 = +ve, 1= -ve). Immediately you have reduced the range of values as one of the bits is reserved for the sign. |

## Data Types (cont)

| | |
|---|---|
| Two's Comp | A improved way of showing negative numbers. If the first bit is a 1, it is taken as the negative version, and all following numbers are added to it. If it is a zero, it behaves just like a negative number.<br><br>To change a normal +ve binary number to twos comp. **Flip the bits, and add 1**<br>E.g.<br>01101010 = 106<br>10010101 (Flip the bits)<br>10010110 = +1<br>10010110 = -106 |

By **0llieC** (0llieC)
cheatography.com/0lliec/

Published 20th February, 2019.
Last updated 12th June, 2017.

## Data Types (cont)

| | |
|---|---|
| Fixed Point Binary | A method of showing floats. The decimal is **fixed** so there is a set amount of integer bits and a set number of fractional bits. The fractional parts follow the same positional numbering (powers of 2) but the negative versions i.e. $2^{-1}$, $2^{-2}$, etc. |

## Data Types (cont)

| | |
|---|---|
| Floating Point Binary | A method of showing binary composed of 2 parts: the mantissa, and the exponent. The mantissa is the actual number, and the exponent is the number of units to move the floating point up or down by.<br><br>10101011 \| 0011<br>Mantissa \| Exponent<br><br>$1.0101011 \times 2^{0011}$<br>$1.0101011 \times 2^{3}$<br>1.0101011<br>→ → →<br>1010.1011 = 10.6875<br><br>The rule for powers:<br>→<br>↓<br>(Decrease the power, per jump right) |

## Data Types (cont)

| | |
|---|---|
| Underflow | When **very** small numbers, and the boundary of what the computer can store is reached. For example, $128^{-1} \times 128^{-1}$ requires 14 bits to be stored. |

By **0llieC** (0llieC)
cheatography.com/0lliec/

Published 20th February, 2019.
Last updated 12th June, 2017.

## Data Types (cont)

| | |
|---|---|
| Overflow | A calculation that results in a number too large to be stored. For example, any numbers past $x10^{100}$ on most calculators |

## Logic Gates

| | |
|---|---|
| Half Adder | Add two single bits, produces an output S, and a carry signal C. It consists of an AND gate (C) and a XOR gate (S) in parallel. |

## Logic Gates (cont)

| | |
|---|---|
| Full Adder | Full adders are a combination of 2+ half adders. Where the C of both half adders, connects to a OR gate ($C_{out}$) and the sum of one connects as the input of another. Basically treating the first HA as another input. |

By **0llieC** (0llieC)
cheatography.com/0lliec/

Published 20th February, 2019.
Last updated 12th June, 2017.

| Logic Gates (cont) | |
|---|---|
| Basic Flip-Flop | Takes a set and a reset signal. The idea is that the FF stays in one state until the change signal is sent. |

| Logic Gates (cont) | |
|---|---|
| D (Data) Type Flip Flop | Stores the signal it receives if it is enabled. It takes in an extra input D. |

By **0llieC** (0llieC)
cheatography.com/0lliec/

Published 20th February, 2019.
Last updated 12th June, 2017.