GitHub - BrookJeynes/2020DigitalStudyGuide

BrookJeynes

Digital Solutions 2020 Final Exam Guide

Written by Brook Jeynes, Brady Stroud and Baeleigh Harris

Contents

Topic 1

Topic 2

Data Delivery

Topic 1: Digital methods for exchanging data

Encryption

Criteria:

- Recognise and Describe
  - Encryption and authentication strategies appropriate for securing data transmissions and their differences
  - features of symmetric (Data Encryption Standard — DES, Triple DES, AES — Advanced Encryption Standard, Blowfish and Twofish) and assymetric (RSA) encryption algorithms * How data compression, encryption and hashing are used in the storage and transfer of data
- Symbolise, Analyse and Evaluate
  Caesar, Polyalphabetic (e.g. Vigenere and Gronsfield), and one-time pad encryption algorithms

---

Encryption

Encryption is the process of scrambling data so that only the desired party can understand that information. In more technical terms it is the process of converting plain text to cipher text, with only the desired party having the means to convert it back; the key.

```
"Hello"  --->  [encryption]  --->  "SNifgNi+uk0=" (base64-encoded encrypted bytes)
plaintext                          ciphertext
```

Wait, base64? In modern day encryption, we don't just encrypt text - we can encrypt images, files or even entire executable files. This, of course, doesn't just miraculously turn into text; we need an algorithm for that. Base64 attempts to turn seemingly random encrypted binary data (bytes) into readable text, and is distinctive because of its particular alphanumeric/symbolic encoding. Base64 is useful when you need to encode binary data as text, and as such is used heavily on the web.

Compression

Compression is the process used to reduce the storage space a file or program uses, this allowing more files to fit into the same amount of space. This process is especially useful when transferring files over the internet due to larger files taking a larger amount of time to transfer. Compression is pretty simple when put into practice.

By removing all duplicates (redundancies) and replacing them with 'pointers' to where the data can be found, you reduce the amount of space needed. Pointers (for reference) are usually numbers that tell the computer where to find something in memory, or in a file (offset). For example, a pointer to get from position 512 onward in a text file looks like this: 512 (makes sense, right?).

When decompressing, the computer simply looks up all pointers and pieces back together the data that was the uncompressed file.

Hashing Hashing is the process in which an input is turned into a fixed sized value. Hashing, unlike encryption, has an output that cannot be reversed to form the plaintext, as there is no key involved. This means that if a malicious actor gained access to the hashed database they could only gain access to the 'hash', which cannot be reversed to gain the sensitive data it represents; like passwords or PINs. However, someone with the plaintext already at their disposal can hash it and check it against the stored hash to gain access; of course, both hashes must be identical for the check to succeed. This means that an attacker must 'brute-force' (go through every single combination) the hash in order to create a matching plaintext; which is inefficient, requires massive amounts of computing power and is ncredibly time-consuming.

Examples of hashes:

Algorithm	Plaintext	Hash
MD5	hello	5d41402abc4b2a76b9719d911017c592
SHA-1	goodbye	3c8ec4874488f6090a157b014ce3397ca8e06d4f
SHA256	greetings	7dd4f2f077e449b47215359e8020c0b6c81e184d2c614486246cb8f70cac7a70
SHA512	P@ssW0rd.1	cbe1cbbf03d4fbeb1941ba8cf77bc41b1844aafa30eba064c9147977939a1851728276886cde0d9520fe1fbb0bb1

Generally speaking, the longer the hash ('digest' size), the more secure it is. SHA512 is one of the most efficient and secure algorithms, standardising it as the 'go-to' algorithm for many. MD5, however, has long since been deprecated and is not recommended for use.

In modern times, there are two main types of cryptographic algorithms used to protect raw data during transfers: symmetric and asymmetric encryption.

Symmetric Key Encryption

Symmetric key encryption, also known as a symmetric algorithm, is a type of encryption that uses one key to encrypt and decrypt data, a secret key, a public key and a private key. *"Keys are random bits that are used by the algorithm to transform the material into its encoded format and back to plain text." - ("Encryption 101", 2020)*. Some advantages to using symmetric key encryption include its encryption speed and efficiency for large projects with disadvantages consisting of its need to keep the secret key, this can become tricky when dealing with multiple locations.

Symmetric Key Ciphers

Cipher	Description	Reference
DES	A 64-bit block cipher that uses a 56-bit key. The block size is always constant, and data is 'padded' with 'dummy data' to ensure it meets the correct block size.	(BRH Media, 2020)
Triple DES (3DES)	Uses a 168-bit key (3x the size of DES). The message is encrypted via 48 transform rounds, with the key being transformed at the end of each round to form a 'round key'. Round keys encrypt the next round.	(BRH Media, 2020)
AES	A variant of [the] Rijndael [block cipher], with a fixed block size of 128 bits, and a key size of 128, 192, or 256 bits.	(Advanced Encryption Standard, 2020)
Twofish	A symmetric block cipher which operates on 128 bit blocks and employs 16 rounds with key lengths up to 256 bits.	-
Blowfish	A symmetric block cipher which operates on 64 bit blocks and employs 16 rounds with key lengths up to 448 bits and uses large key-dependant S-boxes [S-box: *'a basic component of symmetric key algorithms which performs substitution'* - ('S-box', 2020)].	-

Asymmetric Key Encryption

Asymmetric key encryption, also known as an asymmetric algorithm, is a type of encryption that uses two separate keys, with one being used to encrypt and the other to decrypt data. The key pair being referenced as a public key and private key. The public key is used to send the message and the private key being the one to decrypt said message. Some advantages to using asymmetric key encryption include its encryption extended functionality and its scalability for larger projects with its main disadvantage being the speed of the algorithm.

Asymmetric Key Ciphers
"In RSA, the public key is generated by multiplying two large prime numbers p and q together, and the private key is generated through a different process involving p and q. A user can then distribute his public key pq, and anyone wishing to send the user a message would encrypt their message using the public key... When the user receives the encrypted message, they decrypt it using the private key and can read the original text." - (Katz et al., 2020)

Caesar Cipher

The Caesar Cipher is one of the earliest known ciphers to have been invented, it is known as a substitution cipher. It works in such a way where each letter in the message is shifted a certain number of places down the alphabet. For example, for a shift of 1 character:

PlainText	*Transform*	CipherText
"Hello"	*[encryption]*	"Ifmmp"

Polyalphabetic

A polyalphabetic cipher was the first main solution at a problem that had plagued ciphers for a while, frequency analysis. Not all characters in the alphabet are made equally and its because of this people were able to start analysing encrypted text looking for which characters appeared the most. This allowed them to map ciphered characters to uncyphered characters based on their frequency rating.

An example of this is that the character "e" is the most common character in English so *"if 'e' has been encrypted to 'X', then every 'X' was an 'e'. Hence, the most common letter in the ciphertext should be 'X'." - (Rodriguez-Clark, 2020)*

Polyalphabetic ciphers incorporates multiple ciphers in one so a single plain text character will not always be the same cipher character.

Monoalphabetic:

PlainText	*Transform*	CipherText
"Hello"	*[encryption]*	"Ifmmp"

Polyalphabetic:

PlainText	*Transform*	CipherText
"Hello"	*[encryption]*	"Ifgmp"

Polyalphabetic Ciphers

- **Vigenere**: "The Vigenère cipher is a method of encrypting alphabetic text by using a series of interwoven Caesar ciphers, based on the letters of a keyword." - ("Vigenère cipher", 2020). Example:

PlainText	Key	CipherText
"Hello"	"secret"	"Zincs"

- **Gronsfield**: "The Gronsfeld cipher is essentially a Vigenere cipher, but uses numbers instead of letters. So, a Gronsfield key of 0123 is the same as a Vigenere key of ABCD." - ("Gronsfeld Cipher", 2020). Example:

PlainText	Key	CipherText
"Hello"	"0123"	"HFNOO"

Visual Communication

** Criteria: * Recognise and Describe * How usability principles are used to inform solution development * How the elements and principles of visual communication inform user interface development* - - -

Usability Principles

There are seven main usability principles that will be focused on, hierarchy, harmony, contrast, repetition, alignment, proximity and balance. The combination of these principles help the users navigate though a webpage with ease giving them the best experience and allowing them to know how the webpage is meant to be used. These principles help visually and practically for the user with the use of colour design all the way to easy navigation for the blind (using screen readers). There are also usability elements, space, line, colour, shape, texture, tone, form, proportion and scale which aid the usability principles.

Usability Principles

- Hierachy: "This principle can be used to highlight the importance of particular content and features, encouraging users to respond to important elements. When designing a web page, it can be helpful to keep in mind the natural tendency of users to assign greater importance to content at the top left of the screen, and less importance as they move from the top down and from left to right." - (School Resources, 2020)
- Harmony: "Harmony offers an interpretation of proximity to ensure components as a whole provide valuable meaning and are complementary across the interface. In a data driven context, it is not always harmonious to place certain datasets with others. Sometimes it is better to place datasets on separate screens to avoid confusion or otherwise improve the experience." - (School Resources, 2020)
- Contrast: "Contrast is what you see when you compare things that are different. Humans are wired to notice differences, and the clever use of contrast can support powerful interactions. Contrast can be achieved in simple ways by utilising elements such as colour and space, and in conjunction with other principles such as proportion and scale." - (School Resources, 2020)
- Repetition: "The principle of repetition provides predictability through the reusability of elements, and is commonly used in data-driven solutions. The repeated elements could be page constructs, sections or product layouts. This allows users to learn the user environment and predict where they will be able to source information they require." - (School Resources, 2020)
- Alignment: "Alignment is a principle of design that is conducive to how users infer information from a layout. Alignment of images, text and objects provides structure, eliminates potential haphazardness and allows the user to effectively scan information." - (School Resources, 2020)
- Proximity: "When presented with information, users will mentally group together elements that are close to each other in a space. It is suggested that many users 'clump' these into a summarised object. It is important to keep this in mind when developing a data-rich user experience. When data is placed close to other data, a relationship is often inferred based on a user's understanding of the logical layout of data. Changing the physical proximity of the data changes its potential meaning and how a user interprets it. This is an important consideration in delivering meaningful data to a user." - (School Resources, 2020)
- Balance: "Balance in an interface refers to a sense of equilibrium and symmetry in the eyes of the user, allowing the user to effortlessly interpret the interface. Balance is sometimes informal — meaning it isn't exact, and is more of a general appreciation of perspective. Balance in a web context may mean simply that the breakdown of the page shows symmetry in relation to chunks of detail." - (School Resources, 2020)

Usability Elements

- Space: "Space can be used to support meaning and to zone groups of data. It can be used consistently to develop predictability throughout a user experience. It should not be considered to be just negative or blank. An interface without the use of structured space could be considered by a user to be too busy, confusing and not learnable." - (School Resources, 2020)
- Line: "The use of the line element in interfaces is well defined and is naturally applied in the principles of alignment, proximity and hierarchy. A line can be thought of as a starting place, a marker or trigger to change. A simple horizontal line with textural and tone features within an interface such as a web page can provide separation or encapsulation." - (School Resources, 2020)
- Colour and Tone: "The use of colour plays a significant role in developing data-driven applications and presentation of associated data. Aspects of colour that should be considered include: A: a consistent approach to colour choice throughout the application. This assists in branding and the predictability of the product for users. B: the compatibility of colours for ease of use. Some designers engage the colour wheel to identify compatible colour choices that support contrasting information as well as making the solution more accessible." - (School Resources, 2020)
- Texture: "Texture is a truly artistic element of visual design and refers to the tactile or inferred visual features of an object. Texture can be used to assist in visual or physical improvements in accessibility, or in conjunction with other elements such as form and shapes to give emphasis or distinction. " - (School Resources, 2020)
- Form: "Utilising the element of form within a user experience can give the feeling of depth and a multi-dimensional perspective. This could assist in highlighting components through the use of shadows on the surfaces or faces of objects. Form is usually applied in conjunction with other elements such as tone, texture and colour." - (School Resources, 2020)
- Proportion and Scale: "Concepts such as the 'golden ratio' are used to demonstrate proportion in design work such as interfaces. This is a commonly occurring mathematical ratio that has been discovered to create aesthetically pleasing, natural-looking presentations." - (School Resources, 2020)

```
<br><br>
```

Australian Privacy Act

Criteria:

- *Explain*
  - *Australian Privacy Principles (2014) and ethics applicable to the use of personally identifiable or sensitive data from a digital systems perspective*

Nice PDF that summarises all APP's There are three main privacy principles that the exam will focus on, these consist of APP 1, 6 and 11

- APP 1 (Open and transparent management of personal information): This principle highlights and prescribes efforts that organisations must take to meet the requirements of the Privacy Act and the Australian Privacy Principles in general. It provides advice to organisations about how to clearly communicate what information they collect, how they hold and use it, and who they can potentially disclose it to. This communication should also include ways in which individuals can access, correct or raise concerns about their information. The main communication strategy the principle suggests is the use of an organisational privacy policy. This policy must be accessible and free to all individuals at any time. For example, when accessing systems or exploring an organisational website, a privacy policy is generally an accessible document that is highlighted or mentioned. Adopting services provided by companies such as Apple also directs users back to privacy disclosure statements that they must acknowledge.

- APP 6 (Use or disclosure of personal information): This principle outlines how organisations may use personal information and in what circumstances they are allowed to disclose it. In a general sense, data must not be disclosed for any use other than its primary purpose unless consent is provided or other laws require it to be disclosed. In most circumstances, permission for third-party disclosures is listed among privacy policies that users are asked to agree to before they receive a service. In all other circumstances, information that is either de-identified or not considered personal or sensitive to an individual may be released. For example, if an organisation wanted to share elements of your personal data with a third-party research group, they would have to acknowledge this in their privacy policy or actively seek your consent.

- APP 11 (Security of personal information): This principle outlines the requirement on organisations to take reasonable steps to protect an individual's personal information from misuse, interference, loss or unauthorised access or disclosure. Methods of security employed include technical and human processes such as encryption, access privileges and restrictions on how the information is accessed — for example, processes around how a customer can call an organisation and gain access to their information. Questions that substantiate a user's identity are among the methods used at present. The principle also denotes that actions should be taken to destroy or de-identify data that the organisation no longer needs.

Networking

- *Criteria:*
  Explain
  Network transmission principles, including latency, jitter, guarantee and timeliness of delivery, and protocols relevant to the transmission of data over the internet, e.g. HTTP, HTTPS, FTP, VPN, streaming and broadcasting data packets

So what is a network? A network is essentially a group of computers that are connected, or linked, that are capable of sharing data, recourses and files with each other. This connection can be through many things but is most commonly linked though a cable/s, telephone lines, satellites or radio waves.

> A useful comic to go along with this section can be found here. This comic will cover most of the content explained below in a fun condensed way. It is recommended you still read this section as it goes into more detail about certain topics.

Latency

Usually measured in milliseconds, latency is the time it takes for data or a request to go from the source to the destination. Latency depends on the type of Network or the type of packets being transferred.

Latency can either be measured as the **Round Trip Time (RTT)** or the **Time to First Byte (TTFB)**:

- **RTT** is defined as the amount of time it takes a packet to get from the client to the server and back.
- **TTFB** is the amount of time it takes for the server to receive the first byte of data when the client sends a request.

Jitter

The deviation from true periodicity of a presumably periodic signal, often in relation to a reference clock signal. In clock recovery applications it is called timing jitter. Jitter may be caused by electromagnetic interference and crosstalk with carriers of other signals. Jitter can cause a display monitor to flicker, affect the performance of processors in personal computers, introduce clicks or other undesired effects in audio signals, and cause loss of transmitted data between network devices. The amount of tolerable jitter depends on the affected application.

Protocols

A protocol is is a system of rules that allows two or more entities of a communications system to transmit information via any kind of variation of a physical quantity. The protocol defines the rules, syntax, semantics and synchronisation of communication and possible error recovery methods. Protocols may be implemented by hardware, software, or a combination of both.

List of common protocols

Protocol	Description	Resources
HTTP	HyperText Transfer Protocol is used for distributed, collaborative, hypermedia information systems. HTTP is the foundation of data communication for the World Wide Web.	HTTP and HTML Explained
HTTPS	HyperText Transfer Protocol Secure is an extension of HTTP used for secure communication over networks. HTTPS is encrypted using TLS (Transport Layer Security). You can tell when your browser is communicating over HTTPS by the green lock icon in the address bar of most browsers. Most browsers also display a warning to the user when visiting a site that contains a mixture of encrypted and unencrypted content.	Why do we need HTTPS?
TCP	Transmission Control Protocol is a standard that defines how to establish and maintain a network conversation through which application programs can exchange data.	What is TCP/IP?
IP	Internet Protocol is the principal communications protocol in the Internet protocol suite for relaying datagrams across network boundaries. Its routing function enables internetworking, and essentially establishes the Internet. It encompasses an entire family of protocols and systems that include (but aren't limited to): • TR-069/181 • DNS • DHCP IP has the task of delivering packets from the source host to the destination host solely based on the IP addresses in the packet headers. For this purpose, IP defines packet structures that encapsulate the data to be delivered. It also defines addressing methods that are used to label the datagram with source and destination information.	*See others*
FTP	File Transfer Protocol is used for the transfer of computer files between a client and server on a computer network. FTP users may authenticate themselves with a clear-text sign-in protocol, normally in the form of a username and password, but can connect anonymously if the server is configured to allow it. For secure transmission that protects the username and password, and encrypts the content, FTP is often secured with SSL/TLS (FTPS). Setting up an FTP control connection is quite slow due to the round-trip delays of sending all of the required commands and awaiting responses, so it is customary to bring up a control connection and hold it open for multiple file transfers rather than drop and re-establish the session afresh each time. In contrast, HTTP originally dropped the connection after each transfer because doing so was so cheap.	How FTP Works
VPN	A Virtual Private Network extends a private network across a public network and enables users to send and receive data across shared or public networks as if their computing devices were directly connected to the private network. Applications running across a VPN may therefore benefit from the functionality, security, and management of the private network. Encryption is a common, although not an inherent, part of a VPN connection.	How do VPNs work?

Methods of data exchange

_Criteria:

- Explain
  Methods for data exchange used to transfer data across networked systems including REST, JSON and XML
- Symbolise and Explain
  - Secure data transmission techniques and processes, including the use of encryption, decryption, authentication, hashing and checksums
  - How data compression, encryption and hashing are used in the storage and transfer of data
- Describe
  Data using appropriate naming conventions, data formats and structures

_

REST API's

What is a REST API?

> **R e**presentational **S**tate **T**ransfer
>
> **A**pplication **P**rogramming **I**nterface

Let's say you're trying to find videos about Batman on Youtube. You open up Youtube, type "Batman" into a search field, hit enter, and you see a list of videos about Batman. A REST API works in a similar way. You search for something, and you get a list of results back from the service you're requesting from.

An **API** is a set of rules that allow programs to talk to each other. The developer creates the API on the server and allows the client to talk to it.

REST determines how the API looks like. It is a set of rules that developers follow when they create their API. One of these rules states that you should be able to get a piece of data (called a resource) when you link to a specific URL.

Each URL is called a request while the data sent back to you is called a response.

The Anatomy Of A Request

It's important to know that a request is made up of four things:

- The endpoint
- The method
- The headers
- The data (or body)

The **endpoint** (or route) is the url you request for. It follows this structure:

`root-endpoint/?`

The root-endpoint is the starting point of the API you're requesting from. The root-endpoint of Github's API is https://api.github.com while the root-endpoint Twitter's API is https://api.twitter.com. The path determines the resource you're requesting for. Think of it like an automatic answering machine that asks you to press 1 for a service, press 2 for another service, 3 for yet another service and so on.

You can access paths just like you can link to parts of a website. To understand what paths are available to you, you need to look through the API documentation. For example, let's say you want to get a list of repositories by a certain user through Github's API. The docs tells you to use the the following path to do so: `/users/:username/repos`

Any colons (`:`) on a path denotes a variable. You should replace these values with actual values of when you send your request. In this case, you should replace `:username` with the actual username of the user you're searching for. If I'm searching for my Github account, I'll replace `:username` with `bradystroud` .

The endpoint to get a list of my repos on Github is this: `https://api.github.com/users/bradystroud/repos`

The final part of an endpoint is **query parameters**. Technically, query parameters are not part of the REST architecture, but you'll see lots of APIs use them. So, to help you completely understand how to read and use API's we're also going to talk about them. Query parameters give you the option to modify your request with key-value pairs. They always begin with a question mark (`?`). Each parameter pair is then separated with an ampersand (&), like this:

`?query1=value1&query2=value2`

JSON

REST API responses are usually in JSON (**J**ava**S**cript **O**bject **N**otation)

JSON Syntax Rules

- Data is in name/value pairs
- Data is separated by commas
- Curly braces hold objects
- Square brackets hold arrays

JSON **data** is written as name/value pairs A name/value pair consists of a field name (in double quotes), followed by a colon, followed by a value:

`"firstName":"John"`

JSON **objects** are written inside curly braces. Objects can contain multiple name/value pairs:

`{"firstName":"John", "lastName":"Doe"}`

JSON arrays are written inside square brackets. An array can contain objects:

```
"employees":[
    {"firstName":"John", "lastName":"Doe"},
    {"firstName":"Anna", "lastName":"Smith"},
    {"firstName":"Peter", "lastName":"Jones"}
  ]
```

In the example above, the object `"employees"` is an array. It contains three objects.

Each object is a record of a person (with a first name and a last name).

Here is another example of JSON:

```
{
  "first name": "John",
  "last name": "Smith",
  "age": 25,
  "address": {
    "street address": "21 2nd Street",
    "city": "New York",
    "state": "NY",
    "postal code": "10021"
  },
  "phone numbers": [
    {
      "type": "home",
      "number": "212 555-1234"
    },
    {
      "type": "fax",
      "number": "646 555-4567"
    }
  ],
  "sex": {
    "type": "male"
  }
}
```

XML

```
<employee>
  <age>25</age>
  <address>
    <city>New York</city>
    <state>NY</state>
    <street_address>21 2nd Street</street_address>
    <postal_code>10021</postal_code>
  </address>
  <sex>
    <type>male</type>
  </sex>
  <first_name>John</first_name>
  <last_name>Smith</last_name>
  <phone_numbers>
    <type>home</type>
    <number>212 555-1234</number>
  </phone_numbers>
  <phone_numbers>
    <type>fax</type>
    <number>646 555-4567</number>
  </phone_numbers>
</employee>
```

Similarities and Differences

Similarities:

- Both JSON and XML can be used to receive data from a web server.
- Both JSON and XML are "self describing" (human readable)
- Both JSON and XML are hierarchical (values within values)
- Both JSON and XML can be parsed and used by lots of programming languages

Differences:

- JSON doesn't use end tags
- JSON is shorter
- JSON is quicker to read and write
- JSON can use arrays

Why JSON is Better:

Using XML

- Fetch an XML document
- Use the XML DOM to loop through the document
- Extract values and store in variables

Using JSON

- Fetch a JSON string

- Parse the JSON string

  John Smith 25 21 2nd Street New York NY 10021 home 212 555-1234 fax 646 555-4567 male

Sub-Systems

Criteria:
- *Symbolise and Explain*
  - *how application sub-systems, e.g. front end, back end, work together to constitute a solution*

Sub-Systems

In computing terms front-end and back-end refer to the sections that make up a software project. The front-end, or presentation layer, refers to anything the user can see. This generally includes the UI or the buttons and inputs the user interacts with. The back-end, or data access layer, is what runs the front-end of the application. The back-end can take care of any data that needs transferring or it can make the buttons and inputs from the front-end functional. When both combined they make a visual appeasing and functional application for the user to interact with.

Methods of data exchange

Algorithms

Criteria:

- *Symbolise, Analyse and Evaluate*
  Caesar, Polyalphabetic (e.g. Vigenere and Gronsfield), and one-time pad encryption algorithms
- *Symbolise and Explain*
  The basic constructs of an algorithm, including assignment, sequence, selection, condition, iteration and modularisation

Algorithms

An algorithm is a finite sequence of set tasks something, generally a computer, will follow to achieve a task set by the algorithm creator. Any set of tasks can be considered as an algorithm, it doesn't need to be computer related. A set of directions of a recipe to make a cake can be considered as algorithms because they are a list of tasks that achieve an end-goal.

Pseudocode

Pseudocode is an artificial and universal language, or set of guidelines, created as a way to write prototype code that can be read by anyone and translated into any language. Like how a blueprint of a building can be read and used by everyone in the building process pseudocode is an application blueprint able to be read in any programming language. Here is an example of a simple pseudocode document:

```
BEGIN
    SET ranNum = random(1, 10)

    REPEAT
        OUTPUT "Guess the number between 1 and 10: "
        INPUT userNum

        IF userNum > ranNum THEN
            OUTPUT "Lower"
        ELSE
            OUTPUT "Higher"
        ENDIF
    UNTIL userNum = ranNum

    OUTPUT "Correct"
END
```

Pseudocode Standards

Rule	Example
Keywords: Keywords are written in bold capitals.	**IF, THEN, ELSE, OUTPUT, WRITE**

Rule	Example

Indentation: Statements that form part of a repetition loop are indicated by the same amount too indicate that they form a logical grouping. *(Due to formatting issues an indent will be represented by 4 "." in a row, these are not usually written with an indent in pseudocode)*	**IF** condition **THEN**
	....statements
	ENDIF

Ending a repetition loop: The end of repetition loops and IF, THEN, and ELSE statements are explicitly indicated.	**IF** condition **THEN**
	....statements
	ELSE
	...statemen
	ENDIF
	
	WHILE condition
	....statements
	ENDWHILE

Clarity: Pseudocode should clearly indicate what is happening at each step, including formulas of calculations.	`CALCULATE net` is not as clear as `CALCULATE net = gross - tax`
Abbreviate: Use a more abbreviated version in which more memory cells used to store the input are given program-like names.	`INPUT num1` is preferred over `INPUT first number`
Naming Conventions: Use camel case naming conventions for variables, sub-routines, methods and functions.	- newString; - getNewString() - myVariableName;
Modularisation: Pseudocode always starts and ends with the BEGIN and END keywords.	Main Algorithm **BEGIN** ....statements **END** Subroutines **BEGIN** name ....statements **END** name
Variables: Programmers use names without spaces for variables.	`INPUT num1` `SET num1 to 1`
Input, Assign, Output: To input, assign or output values, common words can be used as keywords.	- **INPUT** mark - **WRITE** "The total is" count - **PRINT** x, y - **DISPLAY** name, result - **READ** name from list.txt - **OUTPUT** average
Assignment: Pseudocode should clearly indicate what is happening at each step.	**CALCULATE** net = gross - tax

Iteration: Control structures to provide repetition.	**REPEAT**
	....statements
	UNTIL condition
	
	WHILE condition
	....statements
	ENDWHILE
	
	FOR count = startVal **TO** endVal
	....statements
	NEXT count

Content Source: Digital Solutions 2019 v1.2 Supporting Resource: Representing algorithms with pseudocode

Dataflow

Criteria:

Symbolise
Data flow through a system using data flow diagrams

Dataflow Diagrams

A dataflow diagram is a way to plan and understand the flow of data through a process or system. Using predefined shapes, like rectangles and arrows, inputs, outputs and data routes can all be represented in a neat and accessible way. DFD's can be hand drawn simple level systems up to complex multilevel diagram.

Dataflow Diagram Example


data flow diagram

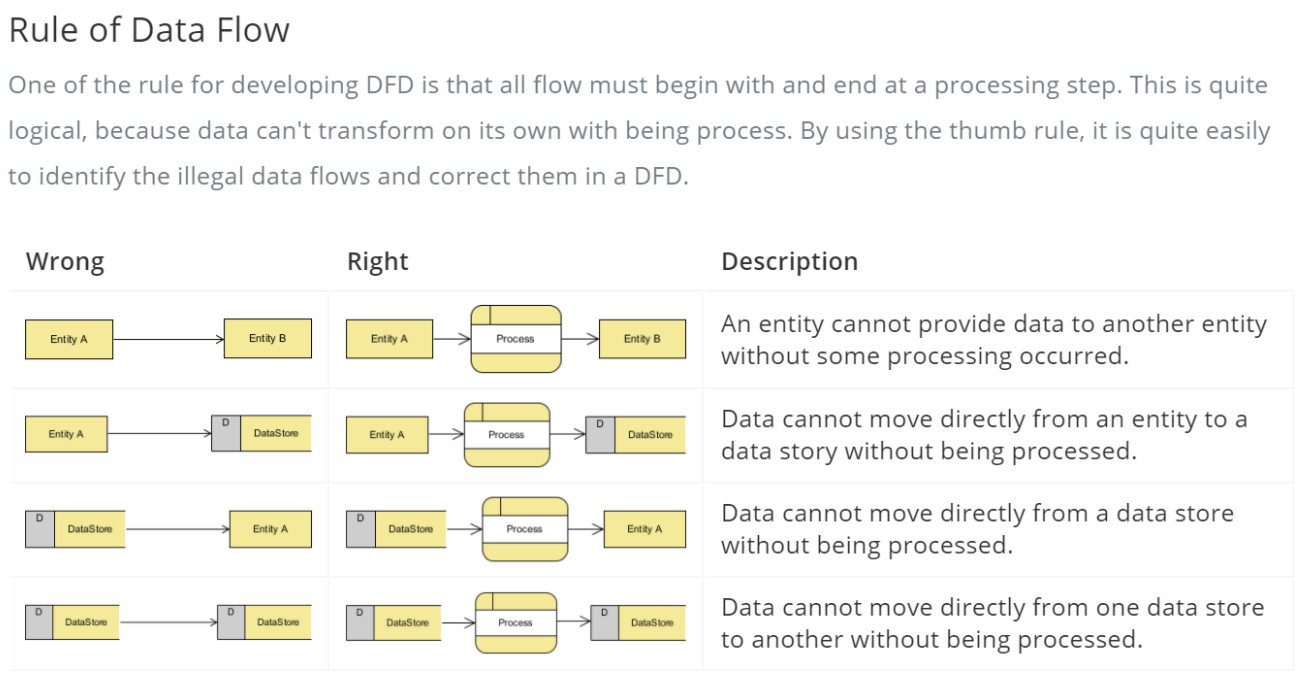
Symbols and Representations

Using any convention's DFD rules or guidelines, the symbols depict the four components of data flow diagrams.

1. **External entity:** an outside system that sends or receives data, communicating with the system being diagrammed. They are the sources and destinations of information entering or leaving the system. They might be an outside organization or person, a computer system or a business system. They are also known as terminators, sources and sinks or actors. They are typically drawn on the edges of the diagram.
2. **Process:** any process that changes the data, producing an output. It might perform computations, or sort data based on logic, or direct the data flow based on business rules. A short label is used to describe the process, such as "Submit payment."
3. **Data store:** files or repositories that hold information for later use, such as a database table or a membership form. Each data store receives a simple label, such as "Orders."
4. **Data flow:** the route that data takes between the external entities, processes and data stores. It portrays the interface between the other components and is shown with arrows, typically labeled with a short data name, like "Billing details."

Content source: "What is a Data Flow Diagram", 2020

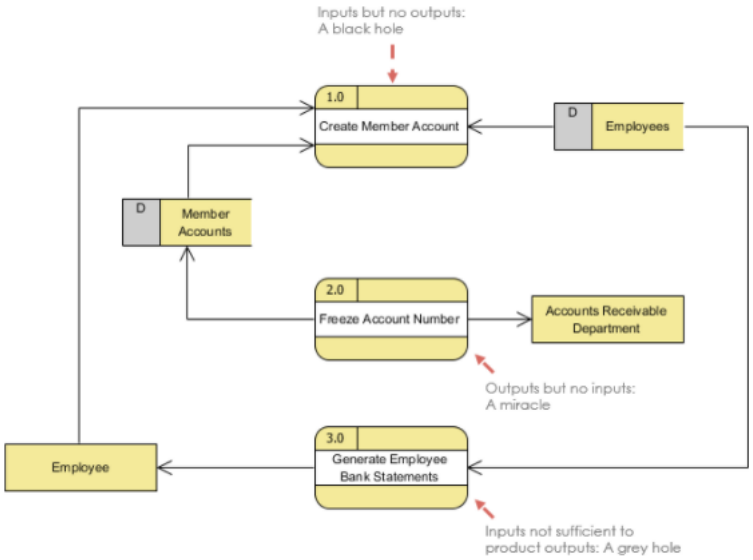
Rules of a Dataflow Diagram



Other frequently-made mistakes in DFD

A second class of DFD mistakes arise when the outputs from one processing step do not match its inputs and they can be classified as:

- Black holes - A processing step may have input flows but no output flows.
- Miracles - A processing step may have output flows but no input flows.
- Grey holes - A processing step may have outputs that are greater than the sum of its inputs



Topic 2: Complex digital data exchange problems and solution requirements

Data Delivery

Criteria:

Analyse Problems and Information to Determine
Factors and risks that affect data security, including confidentiality, integrity and availability, and privacy

- *Analyse, Evaluate and make Refinements to*
  - *Data to ensure completeness, consistency and integrity*

CIA Triad

The websites we use everyday should be secure and allow data to be sent from computer to server without having to worry about an attack or a breach of ones data, one of the main ways a website implements security is with the CIA triad.

> **C**onfidentiality **I**ntegrity **A**vailability

Confidentiality

Confidentiality refers to how a business or organisation keeps its data private and unreachable by any outside organisations. This is generally done by only allowing those who need the file access while preventing unauthorised members access. Confidentiality can be violated for gain in many ways such as a direct attack designed to give the attacker unauthorised access to certain files or data stores. Confidentiality can also be violated unintentionally and by accident by everyday employees using weak passwords or sharing accounts.

Integrity

Integrity refers to ensuring any data, whether in transit or not, is not modified or tampered with in any way, with or without malicious intent. Integrity of data can be checked using methods such as hashing, digital signatures, digital certificates and many various intrusion detection systems.

Availability

Availability refers to the guarantee that the network, site or application will be up and running whenever a user will need to use it. There are many things that could cause this guarantee no fail, simple things that can affect this are power outages, software/system fails or natural disasters. However, this guarantee can be violated on purpose by attackers using various methods to shut down and overload the

system of said attack.