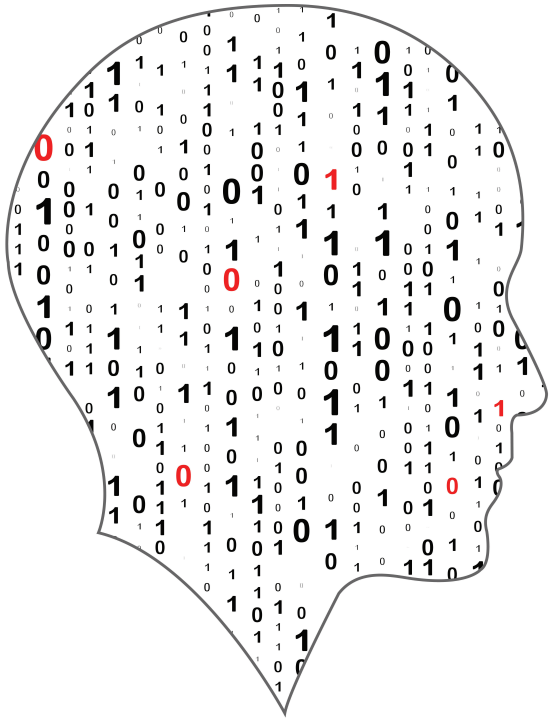
Oide

Tacú leis an bhFoghlaim
Ghairmiúil i measc Ceannairí
Scoile agus Múinteoirí

Supporting the Professional
Learning of School Leaders
and Teachers



LEAVING CERTIFICATE
COMPUTER SCIENCE

National Workshop 3

Professional Learning Booklet

2023 - 2024

Contents

Introduction **3**
- Key Messages 3

Session 1: Computational Thinking **4**
- Xs and O's - a winning Strategy 4
- IQ and Gender 5
- Wing v Denning 6
- Cryptocurrencies - Semi-primes 8

Session 2: Algorithms 1 **9**
- Algorithms and the LCCS Specification 9
- What is an algorithm? 10
- Mean, Median, Mode 12
- Algorithm for Mean, Median, Mode 13
- BREAKOUT TASKS 16

Session 3 – Computer Systems II **19**
- Converting between Number Systems 19
- ASCII to Hexadecimal Table 20
- Hitomezashi Template 21
- Main Components of a Computer - Quizlet Activity 22
- PC Part Picker Activity 23

Session 4: Introduction to Data Analytics ALT 2 **25**
- Data analytics - Reflection Task 25
- Data Science Terminology 25
- Introducing ALT 2 - Group Task 26
- The Data Science Arc 26
- The Data Cleansing Task 27
- ALT 2: Investigate 28

Session 5: Planning and Designing ALT 2 **29**
- ALT 2: Plan 29
- ALT 2: Design 30

Session 6: NCCA Resources, Python libraries & Curriculum Planning **32**
- ALT 2: Create 32
- NCCA Resources 33
- Curriculum Planning 36
- Final Reflection (3-2-1) 37
- Resources 38

Introduction

Key Messages

Leaving Certificate Computer Science aims to develop and foster the learner's creativity and problem-solving, along with their ability to work both independently and collaboratively.

Computing technology presents new ways to address problems and computational thinking is an approach to analyse problems, design, develop and evaluate solutions.

The ALTs provide opportunities for students to develop their theoretical and procedural understanding of the course.

The externally assessed coursework will be based on all learning outcomes, with those of strand 3 being particularly relevant.

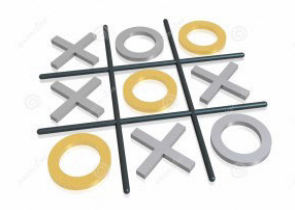
Digital technologies can be used to enhance collaboration, learning and reflection.

Notes

Session 1: Computational Thinking

Develop a winning strategy for Xs and Os, using some of Abstraction? Decomposition? Pattern Recognition? (Monte Carlo?) Algorithm formation? You may want to try it when your opponent goes first.

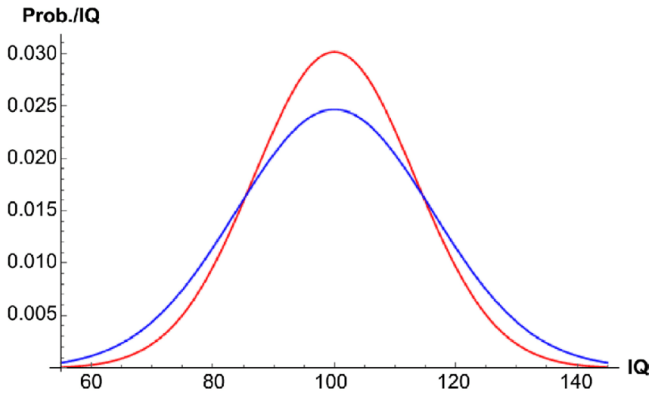
Xs and Os



IQ and Gender

Test the assertion “Females are more intelligent than males”, by considering median, mean, mode, and spread in the graph shown.

Are females more intelligent than males?



Wing v Denning

Computational Thinking is a major element of LCCS. For example,

- Strand 1: Practices and principles, LOs 1.1 – 1.10
- Strand 2: Core Concepts, i.e. Abstraction, Algorithms, Data, and Evaluation and Testing

The purpose of this section is to offer teachers space to build up their own understanding of computational thinking and consider how it might be mediated in the classroom. In order to do this we will explore two opposing viewpoints on computational thinking i.e. those of Peter Denning[1] and Jeanette Wing[2].

tinyurl.com/LCCS-CT-Denning	tinyurl.com/LCCS-CT-Wing

Read both viewpoints (there is space on the next page to record your notes) and consider the following questions:

- *What is computational thinking?*
- *Is Computational Thinking good for everyone?*
- *How does Computational Thinking relate to programming?*
- *How does Computational Thinking relate to other subjects?*
- *How can Computational Thinking be assessed?*
- *Do you think Computational Thinking is best taught or learned?*

[1] Denning, P., Communications of the ACM, Vol 60, 6 (June 2017), 33-39 Remaining Trouble Spots with Computational Thinking

[2] Wing, J., Communications of the ACM, Vol 49, 3 (March 2006), 33-35 Computational Thinking

Wing v Denning

Cryptocurrencies - Semi-primes

A semiprime is a number which only has 2 factors (apart from itself and 1). Finding the factors is often done by a brute force method of finding the first factor by trying every prime number.

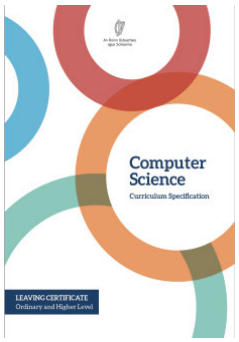
This is a very long process, but it can be made more efficient using some of the pillars of Computational Thinking. Can you formulate a more efficient algorithm to find these factors?

Finding factors of semi-primes

Session 2: Algorithms I

Algorithms and the LCCS specification

What does the LCCS specification say about algorithms?
Use the link or the QR code provided to browse to the LCCS specification.



https://ncca.ie/media/3369/computer-science-final-specification.pdf

Record your findings in the space below. You may use the following questions as a guide.

- *Search for the word 'algorithm'. How many times does it appear in the specification?*
- *What are the core concepts listed in strand 2?*
- *What Learning Outcomes are relevant to algorithms?*
- *Name five algorithms that are mentioned in the specification.*
- *What algorithms relate specifically to ALT2 (Analytics)?*

Notes

What is an algorithm?

A step-by-step procedure for solving a problem or accomplishing some end.[3]

A more formal definition for an algorithm provided by Donald Knuth[4] is as follows:

> *An algorithm is a set of rules for getting a specific output from a specific input. Each step must be so precisely defined that it can be translated into computer language and executed by machine.*

The word *algorithm* itself comes from the Persian polymath *Muḥammad ibn Musā Al-Khwārizmī*, from the region of Khwarazm in what is now Uzbekistan. His book *The Compendious Book*[5] *on Calculation by Completion and Balancing* (c. 820CE) introduced the word algebra and 12th century Latin translations of his textbook on arithmetic introduced Indian numerals and decimal notation to the Western world.

does the LCCS specification say about algorithms?

Some key features of algorithms are…

[3] https://www.merriam-webster.com/dictionary/algorithm

[4] Knuth, D The Art of Computer Programming (Vol. 1, Fundamental Algorithms, 3rd ed.)

[5] Esposito, John L. , ed. (1999) The Oxford History of Islam, Oxford University Press

Think - Pair - Share

Think: Participants spend time in silence writing or thinking about their own ideas.

Pair: Participants turn to the person next to them to discuss their ideas with a partner.

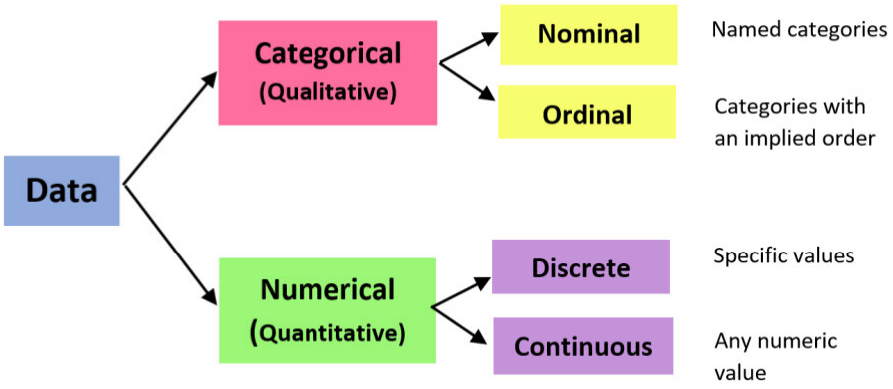
Share: Participants share their answers with another group

does the LCCS specification say about algorithms?

Question	What I thought	What partner thought	What we'll share
Everyday examples of algorithms?			
Why is the study of algorithms important?			

Mean, Mode and Median

As a prerequisite to any data analysis it is both useful and necessary to first understand underlying data type i.e. numeric (continuous vs. discrete) and categorical (ordinal vs. nominal)



Three measures of central tendency - mean, median and mode.

Mean - the sum of all the values divided by the total number of values

Median - the middle value of an ordered set of values i.e. approx 50% higher and 50% lower

Mode - the most commonly occurring value in a distribution

```
# A program to demonstrate the use of some statistics functions
import statistics

# Initialise a list of values
values = [2,3,5,2,4]

# Compute the 3 averages
arithmetic_mean = statistics.mean(values)
median_value = statistics.median(values)
modal_value = statistics.mode(values)

# Display the answers
print("The mean is ", arithmetic_mean)
print("The median and mode are %d and %d" %(median_value, modal_value))
```

When the program is run, the output looks like this:

Algorithm for mean 

```
# Program to find the mean of a list of values
# Version 1

# Calculate and return the mean of all the values in L
def arithmetic_mean(L):

    # set the initial value of total to zero
    total = 0 # running total of values in L

    # Now loop over the list
    for v in L:
        total = total + v # running total

    # Divide by the total by the number of values in L
    return total/5

# PYTHON STARTS EXECUTING FROM HERE ...
# Initialise a list of values
my_list = [18, 27, 15, 13, 22]
# Call the function
my_mean = arithmetic_mean(my_list)
# Display the answer
print("The mean is ", my_mean)
```

tinyurl.com/LCCSmean1

Notes:

Algorithm for Median 

```
# A program to find the median of a list of values
# Version 1

L = [18, 27, 15, 13, 22]

# To find the median we need to sort the list
L.sort() # the values are sorted 'in place'

# The next step is to find the index of the middle value
num_values = len(L)
mid = num_values//2

median = L[mid] # the median is in the middle

# Display the result
print("The median value is: %.2f" %median)
```

tinyurl.com/LCCSmedian1

Notes:

Algorithm for mode 

```
# A program to find the mode of a list of values
# Version 1

# Initialise a list of values
L = [18, 16, 17, 18, 19, 18, 17]

# Build up a list of unique values
unique_values = []
for value in L:
    if value not in unique_values:
        unique_values.append(value)

# Build up a list of frequencies
frequencies = []
for value in unique_values:
    frequency = L.count(value)
    frequencies.append(frequency)

# Find the mode
max_frequency = max(frequencies)
max_frequency_pos = frequencies.index(max_frequency)
mode = unique_values[max_frequency_pos]

print("Mode is", mode)
```

tinyurl.com/LCCSmode1

Notes:

BREAKOUT TASKS

Breakout task 1 - mean

Download the starter code for this task to your Python IDE and save as 'mean1.py'.
The starter code is available from tinyurl.com/LCCSmean1

Follow the instructions in the EXPERIMENT and TASKS FOR BREAKOUT sections.
The EXTENSION EXERCISES can be explored after the workshop

EXPERIMENT ...

1. Check that the program is working (i.e. actual result Vs expected result)
2. Change the values in the list and see what happens
3. What happens if there were more (or less) than 5 values in the list?
4. Does this program work for a list of strings (e.g. ["Mary", "Andy", "Pat"])? Why/why not?

TASKS FOR BREAKOUT

1. Save this program as mean2.py
2. Change the comment at the top of the program to say Version 2
3. Modify the code so that it works for any number of values (not just 5)
   HINT: Use the built-in functions sum and len (see SEC Python reference guide)
4. Modify the code so that it works without a loop

EXTENSION EXERCISES (post-workshop tasks)

1. Modify the code to get the mean of numbers entered by the end user
   HINT: Ask in advance how many numbers they will enter and use a for loop
2. Modify the code to get the mean of numbers entered by the end user but instead of using a for loop, use a while loop and a loop guard to determine when to stop
3. Modify the code to get the mean of numbers read in from a text (or csv) file

Notes:

Breakout task 2 - median

Download the starter code for this task to your Python IDE and save as ‘median1.py’.
The starter code is available from tinyurl.com/LCCSmedian1

Follow the instructions in the EXPERIMENT and TASKS FOR BREAKOUT sections.
The EXTENSION EXERCISES can be explored after the workshop

EXPERIMENT

1. Check that the program is working (i.e. actual result Vs expected result).
2. Change the values in the list and see what happens.
3. What happens if there was an even number of values?
4. Investigate the difference between 'sort()' and 'sorted()'.
   HINT: replace the line *L.sort()* with *sorted(L)* - what effect does this have on the program?
5. Does this program work for a list of strings (e.g. ["Mary", "Andy", "Pat"])? Why/why not?

TASKS FOR BREAKOUT

1. Save this program as median2.py
2. Change the comment at the top of the program to say Version 2
3. Modify the code so that it works for an even number of values

EXTENSION EXERCISES (post-workshop tasks)

1. Incorporate the code into a function (name it calc_median)

Notes:

Breakout task 3 - mode

Download the starter code for this task to your Python IDE and save as 'mode1.py'.
The starter code is available from tinyurl.com/LCCSmode1

Follow the instructions in the EXPERIMENT and TASKS FOR BREAKOUT sections.
The EXTENSION EXERCISES can be explored after the workshop

EXPERIMENT

1. Check that the program is working (i.e. actual result Vs expected result).
2. Change the values in the list and see what happens.
3. What happens if there was a) no mode b) several modes?
4. Does this program work for a list of strings (e.g. ["Mary", "Andy", "Pat", "Mary"])? Why/why not?

TASKS FOR BREAKOUT

1. Save this program as mode2.py
2. Change the comment at the top of the program to say Version 2
3. Modify the code so that it displays the frequency of the mode
4. The statement list(set(L)) returns a list of unique elements in L. Use this information to shorten the code
5. Incorporate the code into a function (name it calc_mode)

EXTENSION EXERCISES (post-workshop tasks)

1. The program works if the list has a single mode. Modify the code so that it
   a) works if there is no mode e.g. L = [18, 16, 17, 21, 19, 22]
   b) works if there are multiple modes e.g. L = [18, 16, 17, 18, 17, 18, 17]

Notes:

Session 3: Computer Systems II

Converting between Number Systems

Hexadecimal (0-15): 0,1,2,3,4,5,6,7,8,9,A,B,C,D,E,F

Decimal (0-10): 0,1,2,3,4,5,6,7,8,9

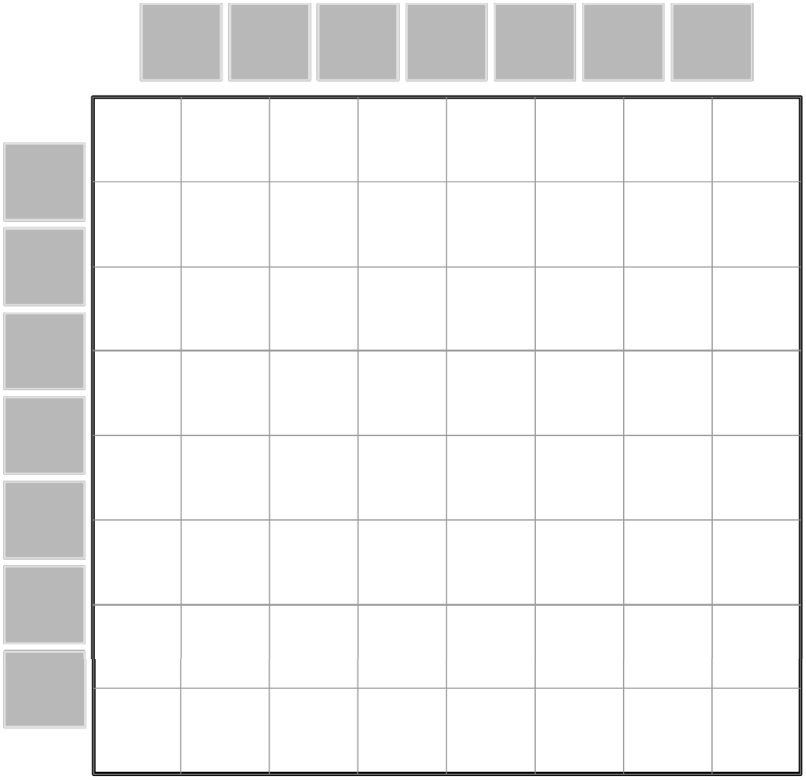
Binary (0-1) 0,1

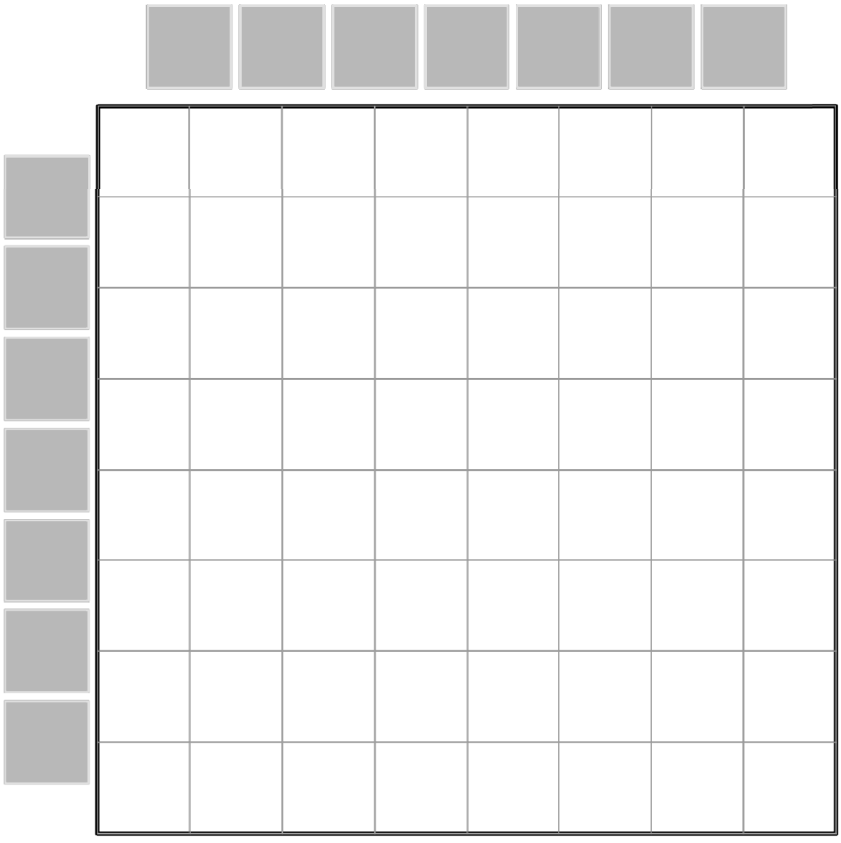
Space for hexadecimal conversion:

ASCII to Hexadecimal Table

Hex	Value	Description	Hex	Value	Description	Hex	Value	Description	Hex	Value	Description
0	NUL	"null" character	10	DLE	data link escape	20	SP	space	30	0	zero
1	SOH	start of header	11	DC1	device control 1 (XON)	21	!	exclamation mark	31	1	one
2	STX	start of text	12	DC2	device control 2	22	"	double quote	32	2	two
3	ETX	end of text	13	DC3	device control 3 (XOFF)	23	#	number sign	33	3	three
4	EOT	end of transmission	14	DC4	device control 4	24	$	dollar sign	34	4	four
5	ENQ	enquiry	15	NAK	negative acknowledgement	25	%	percent	35	5	five
6	ACK	acknowledgment	16	SYN	synchronous idle	26	&	ampersand	36	6	six
7	BEL	bell	17	ETB	end of transmission block	27	'	single quote	37	7	seven
8	BS	backspace	18	CAN	cancel	28	(	left/opening parenthesis	38	8	eight
9	HT	horizontal tab	19	EM	end of medium	29	)	right/closing parenthesis	39	9	nine
0A	LF	line feed	1A	SUB	substitute	2A	*	asterisk	3A	:	colon
0B	VT	vertical tab	1B	ESC	escape	2B	+	plus	3B	;	semicolon
0C	FF	form feed	1C	FS	file separator	2C	,	comma	3C	<	less than
0D	CR	carriage return	1D	GS	group separator	2D	-	minus or dash	3D	=	equality sign
0E	SO	shift out	1E	RS	request to send/record separator	2E	.	dot	3E	>	greater than
0F	SI	shift in	1F	US	unit separator	2F	/	forward slash	3F	?	question mark
Hex	**Value**	**Description**	**Hex**	**Value**	**Description**	**Hex**	**Value**	**Description**	**Hex**	**Value**	**Description**
40	@	"at" symbol	50	P	Capital P	60	`	Grave / accent	70	p	Small p
41	A	Capital A	51	Q	Capital Q	61	a	Small a	71	q	Small q
42	B	Capital B	52	R	Capital R	62	b	Small b	72	r	Small r
43	C	Capital C	53	S	Capital S	63	c	Small c	73	s	Small s
44	D	Capital D	54	T	Capital T	64	d	Small d	74	t	Small t
45	E	Capital E	55	U	Capital U	65	e	Small e	75	u	Small u
46	F	Capital F	56	V	Capital V	66	f	Small f	76	v	Small v
47	G	Capital G	57	W	Capital W	67	g	Small g	77	w	Small w
48	H	Capital H	58	X	Capital X	68	h	Small h	78	x	Small x
49	I	Capital I	59	Y	Capital Y	69	i	Small i	79	y	Small y
4A	J	Capital J	5A	Z	Capital Z	6A	j	Small j	7A	z	Small z
4B	K	Capital K	5B	[	left/opening bracket	6B	k	Small k	7B	{	left/opening brace
4C	L	Capital L	5C	\	back slash	6C	l	Small l	7C	|	vertical bar
4D	M	Capital M	5D	]	right/closing bracket	6D	m	Small m	7D	}	right/closing brace
4E	N	Capital N	5E	^	caret/circumflex	6E	n	Small n	7E	~	tilde
4F	O	Capital O	5F	_	underscore	6F	o	Small o	7F	DEL	delete

Hitomezashi Templates





Main Components of a Computer - Quizlet Activity

Match the terms with their definitions

I/O devices

Storage

Can be likened to being the engine or brain of the computer. It performs the vast majority of the computational tasks in a computer. It's speed depends on a number of factors including clock speed and the number of cores

A type of CPU designed specifically for performing computational tasks relating to the creation/manipulation of images. One way in which this is accomplished is by being highly parallel in nature - they can perform many small computations simultaneously, rather than fewer, larger ones as on a CPU.

Central Processing Unit (CPU)

A Printed Circuit Board (PCB) that acts as the central hub of the computer. All devices and components are connected to it and all communication between devices is done through it.

Graphics Processing Unit (GPU)

Motherboard

Memory

Random Access Memory (RAM) is where the computer will temporarily store information required to complete tasks or keep software running. RAM is accessed/written faster than a disk drive. More RAM means that more and larger files can be stored and accessed ay any one time, leading to more efficient and quicker performance. Anything stored in RAM will erase when the power is turned off.

Any components/peripherals which enable interaction between the computer and the user, e.g. disk drives, monitors, keyboard, etc.

Either on Hard Disk Drives (HDD) or Solid State Drives (SSD) - where the Operating System, software, and files are stored and accessed by other components, as required. HDDs are cheap and can hold large amounts of data. SSDs are faster but currently more expensive for larger sizes.

PC Part Picker Activity

Build a computer for your respective customer.

What does your customer need from their computer?

What hardware is most important for this?

How should you spend your money?

Computer Specification:

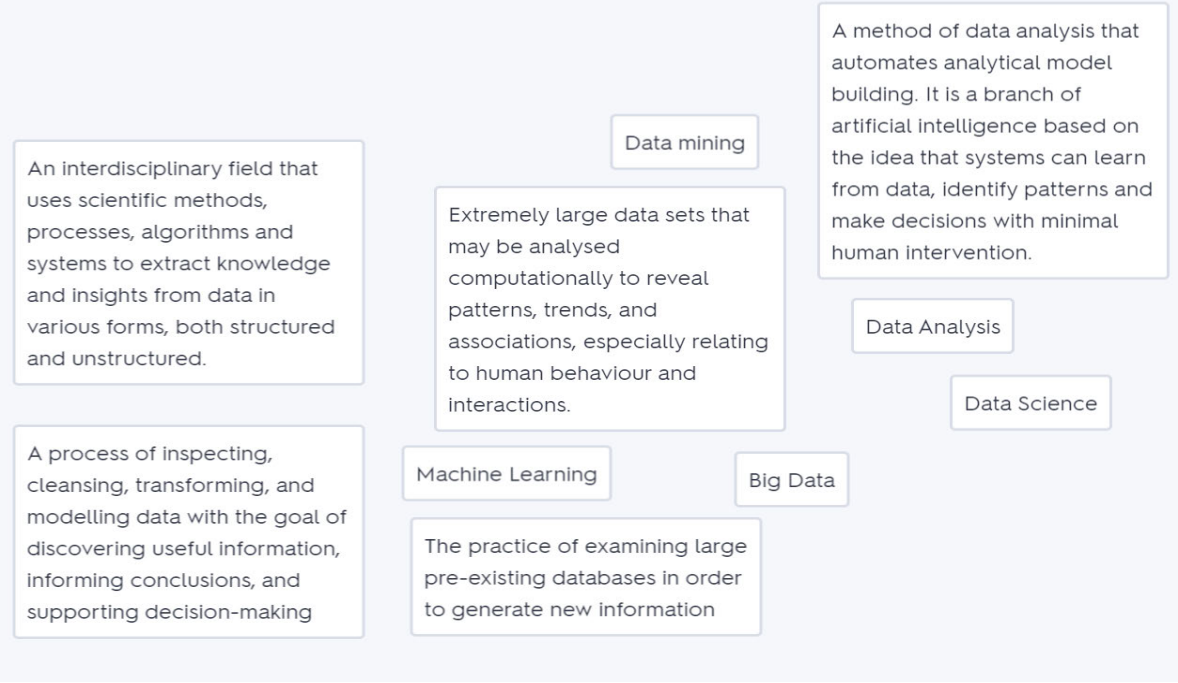
Session 4: Introduction to Data Analytics and ALT 2

Data analytics - reflection task

What words do you associate with data analytics?

Mentimeter

Data science terminology - Quizlet matching activity:



Introducing ALT 2 - group task

Discuss:

1. **What prior knowledge will students have that is relevant to ALT2?**
2. **What may challenge students in dealing with ALT2?**
3. **What approach could you take to introduce ALT2 to your students and support their progress?**

The Data Science Arc

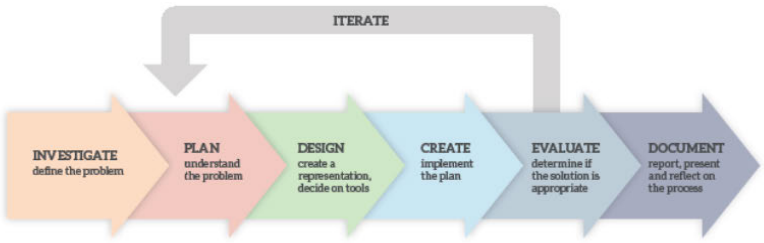


Figure 3: Overview of a design process

Data cleansing task

The data set below shows the raw data collected from the result of a 100m school race.

Surname	Gender	Age	Time
Murphy	M	17	13,12
Ogene	M	16	12.14
Ogene	M	16	12.14
Mc Intyre	F.	17	12.87
Lopez	F	-18	14.01
	F	17	1 329
McCarthy	M	77	13.65
Ó Brádaigh	f	16	13.09

Identify and list any problems associated with the data in the above table.

ALT 2: Investigate

World Happiness Report 2015-2019

https://www.kaggle.com/unsdsn/world-happiness

Spotify Chart Data 2014-2022

https://www.kaggle.com/datasets/jfreyberg/spotify-chart-data

FIFA World Cup 2022

https://www.kaggle.com/datasets/die9origephit/fifa-world-cup-2022-complete-dataset

Significant Earthquakes 1965-2022

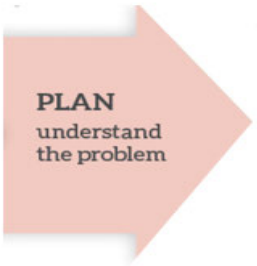
https://www.kaggle.com/usgs/earthquake-database

In your groups, brainstorm possible hypotheses for your dataset.
Aim for as many ideas as you can.

Investigate

Session 5: Planning and Designing ALT

ALT 2: Plan



Choose one hypothesis and dissect your idea.

You may consider the following prompts to help you:

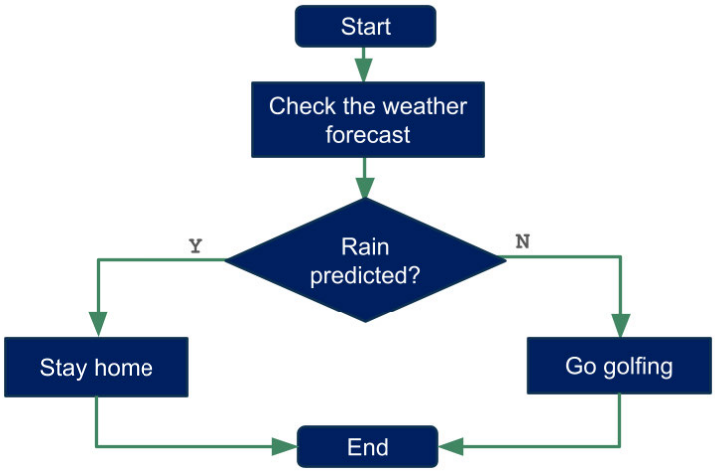
- What does your project do/not do?
- Aims? Any limitations?
- Who are the end users?
- What are the tools/materials required?
- What are the roles and responsibilities?
- Does your project cover all the LOs for this ALT?
- Are there any ethical issues?

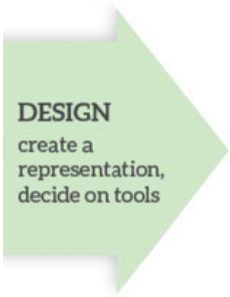
Plan

ALT 2: Design

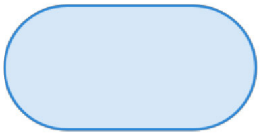
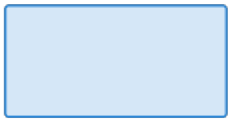
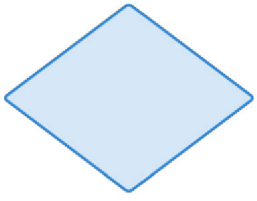
Pseudo-code

```
program start
check weather forecast

if rain predicted
    Stay home
else
    Go golfing
end if

program end
```





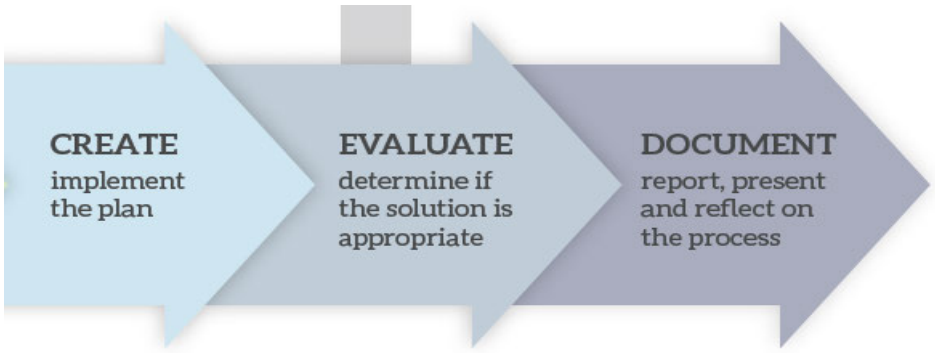
Flowcharts

Symbol	Name	Function
		
		
		
		
		

Design

Session 6: NCCA Resources, Python Libraries & Curriculum Planning

ALT 2: Create



Notes:

NCCA Resources

Use the prompts below to record your thoughts *as the demonstration is being presented.*

How is this demonstration/example extending my thinking in relation to ALT2?

What questions do I have in relation to the demonstration?

What ideas has the demonstration given me that I could use to support my students?

1. Browse to the link below and download one of the sample ALT2 resources.
   tinyurl.com/LCCS-NCCA-ALT2-Examples

2. Unpack the resource and run it from your device

Checklist	Tick (Yes/No)
I know where the samples are	
I know how to download and browse each example	
I understand what I need to do to run the sample locally (in my own local environment)	
I can run the sample locally (in my own local environment)	

3. Browse through the resource and use the space provided on the next page to record your notes about the sample. Explain how the sample fits the Data Science Arc. The prompts may be helpful.
   - *What is the name of the sample?*
   - *What hypothesis is the sample designed to test?*
   - *What data source(s) does the sample use? Where does the data come from?*
   - *What is the output and how is it interpreted in the sample?*
   - *Describe how the algorithm works and identify what programming constructs a student would need to know in order to understand it.*
   - *How do you support students to engage with this process? Consider the Learning Outcomes that could be referenced.*
   - *How might this example be modified/enhanced?*
   - *What new ideas has this example given me that I could try with my own students?*
   - *How does this example extend my thinking in relation to ALT2?*
   - *What ideas has this demonstration given me that I could use to support my own students?*
   - *What hypotheses might students devise that relate to this example.*

Notes:

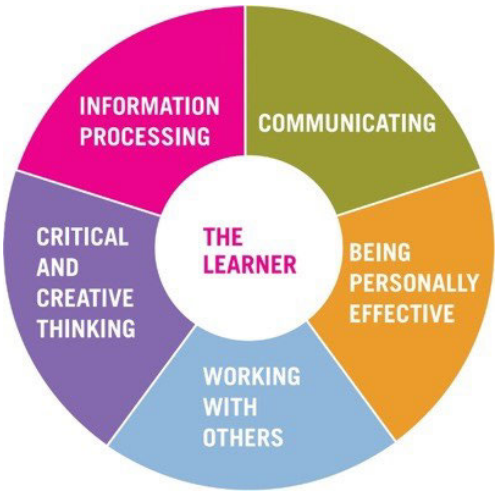
Curriculum planning

Discuss your next steps in relation to curriculum planning.

Focus on ALT 2. Remember to teach the LOs through the lens of the ALTs - there are numerous ways to achieve this.

Consider topics, LOs, build up to ALT2, ALT2, equipment, resources, assessment, differentiation, etc.

Consider the key skills of the senior cycle.



Notes on Curriculum Planning

Final Reflection (3-2-1)

Complete the 3-2-1 reflection with regard to the LCCS specification

List 3 things you learned

List 2 things you'd like to learn more about

One question you still have

Resources

Barr, V. and Stephenson, C. (2011) 'Bringing computational thinking to K-12: what is Involved and what is the role of the computer science education community?', ACM Inroads, 2(1), pp. 48–54. Available at: https://doi.org/10.1145/1929887.1929905.

Denning, P.J. (2017) 'Remaining trouble spots with computational thinking', Communications of the ACM, 60(6), pp. 33–39. Available at: https://doi.org/10.1145/2998438.

Wing, J.M. (2006) 'Computational Thinking', Communications of the ACM, 49(3), pp. 33–35. Available at: https://doi.org/10.1145/1118178.1118215.

Videos

Data Analytics

- What is Big Data? https://www.youtube.com/watch?v=eVSfJhssXUA
- "The Numbers Game: How Data is ChangingFootball" https://www.youtube.com/watch?v=lLcXH_4rwr4
- "The best stats you've ever seen", Hans Rosling https://www.ted.com/talks/hans_rosling_the_best_stats_you_ve_ever_seen
- "Global population growth, box by box", Hans Rosling https://www.ted.com/talks/hans_rosling_global_population_growth_box_by_box

Datasets

- Searchable repository of user-generated datasets (and data challenges) https://www.kaggle.com/
- Central Statistics Office Databases https://www.cso.ie/en/databases/
- Tableau https://public.tableau.com/app/resources/sample-data?qt-overview_resources=1#qt-overview_resources

ALT2 Resources

- ALT2 NCCA resources - https://www.curriculumonline.ie/Senior-cycle/Senior-Cycle-Subjects/Computer-Science/CS-Support-for-Teaching-and-Learning/Support-Material-for-Teaching-and-Learning/2-ALT-Resources/ALT2-Support/
- Python files https://github.com/pdst-lccs/Phase4_NW3 https://github.com/pdst-lccs/NW3-ALT2AlgDemos
- CompSci https://www.compsci.ie/

Notes:

Oide
Tacú leis an bhFoghlaim Ghairmiúil i measc Ceannairí Scoile agus Múinteoirí
Supporting the Professional Learning of School Leaders and Teachers