



Oide

Tacú leis an bhFoghlaim
Ghairmiúil i measc Ceannairí
Scoile agus Múinteoirí

Supporting the Professional
Learning of School Leaders
and Teachers

National Workshop 4

Session 1

Computers and Society I





Key



Booklet



Individual Activity



Group Activity



Reflection



Group Feedback



Key Messages for NW4



There are many ways to use the LCCS specification.



The study of Computers and Society is one of the overarching principles of LCCS.



LCCS can be mediated through a constructivist pedagogical approach.

ALTs

ALTs provide an opportunity to teach theoretical aspects of LCCS.



Critical reflection will be a central component of the student experience and the LCCS teacher's PD journey.

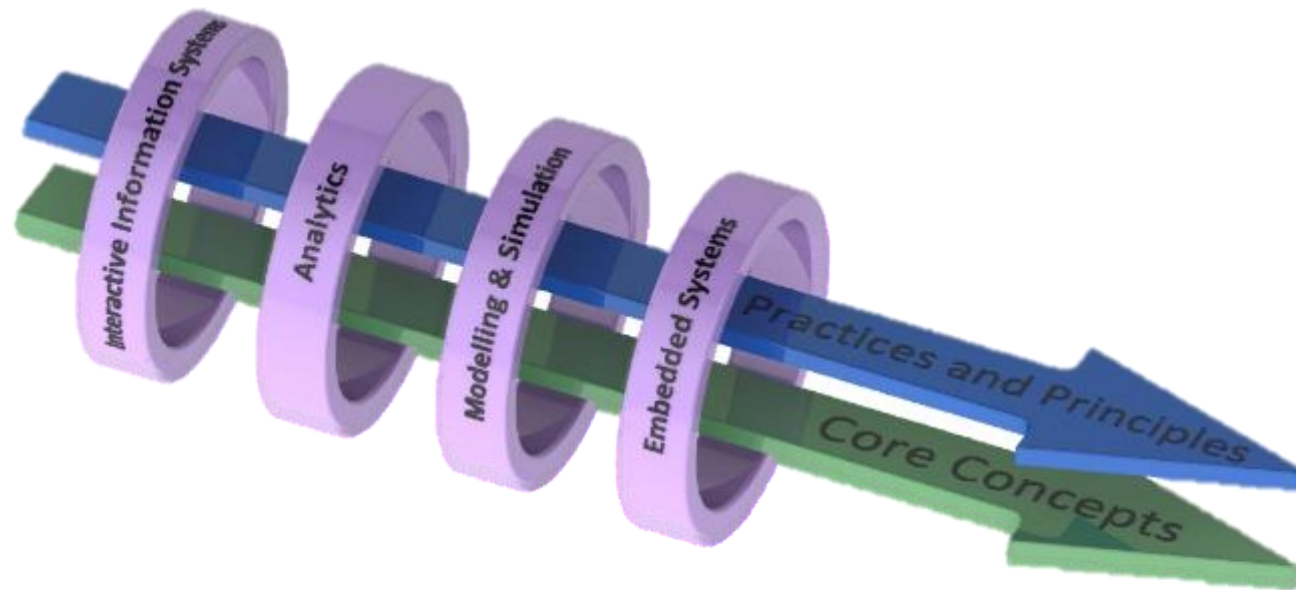


Digital technologies can be used to enhance collaboration, learning and reflection.



LCCS Interwoven

The four applied learning tasks explore the four following contexts:



Key to remember: *Explore and teach the LOs through the lens of ALTs.*



Recap on National Workshop 3

Computer Systems	ASCII and Number Systems
Computational Thinking	PC part picker activity
Unplugged Activities	Hitomezashi Stitching
Introduction to Algorithms	Hypothesis
Algorithms for ALT2 (mean, median and mode)	Python Libraries for ALT2
ALT2 Analytics	ALT2 Investigate, Plan, Design
NCCA Resources for ALT2	Curriculum Planning

Session Overview



Section 1	Introduction to computers and society
Section 2	Introducing <i>Stimulate a debate</i>
Section 3	Group activity (<i>Stimulate a debate</i>)



By the end of this session participants will have...

- gained a deeper understanding of *Strand 1: Computers and Society* (including the LOs)
- explored some strategies and ideas on teaching and learning this section of the course
- gained a deeper understanding of how to *Stimulate a Debate* in a classroom using the four- step process
- experienced the *Stimulate a Debate* via a group activity and, in doing so, appreciate the benefits of this approach
- reflected on how to teach aspects of this section of the course via group activity debrief



Strand 1: Practices and principles

Strand 1: Practices and principles	Strand 2: Core concepts	Strand 3: Computer science in practice
<ul style="list-style-type: none">▶ Computers and society▶ Computational thinking▶ Design and development	<ul style="list-style-type: none">▶ Abstraction▶ Algorithms▶ Computer systems▶ Data▶ Evaluation/Testing	<ul style="list-style-type: none">▶ Applied learning task 1<ul style="list-style-type: none">- Interactive information systems▶ Applied learning task 2 - Analytics▶ Applied learning task 3<ul style="list-style-type: none">- Modelling and simulation▶ Applied learning task 4<ul style="list-style-type: none">- Embedded systems

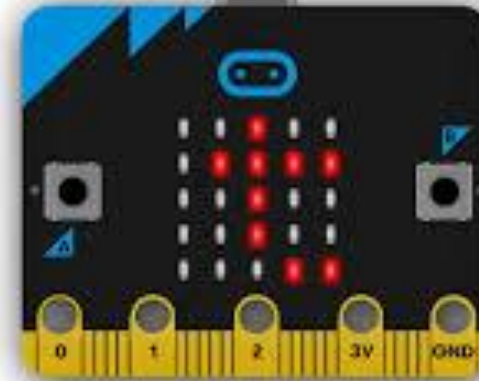
“The practices and principles of computer science describe the behaviours and ways of thinking that computationally-literate students use to fully engage in a data-rich and interconnected world.”

“Studying the role of computers in society will enhance students’ attitudes towards computer science and make it more meaningful and relevant.”



LCCS Interwoven

S3 - ALT 4



S1 - Computers and Society

S1 - Designing and Developing

S1 - Computational thinking

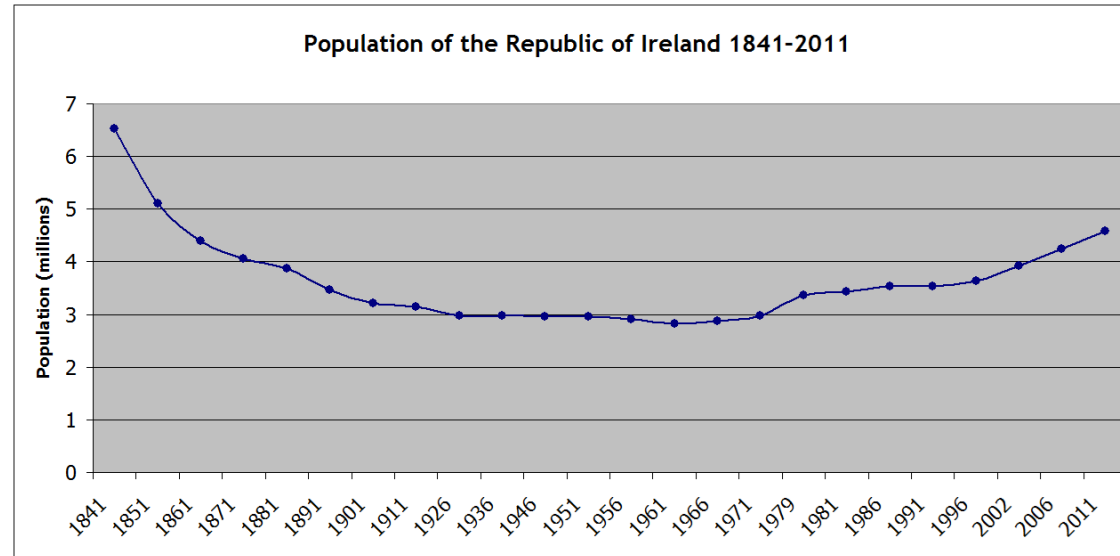
S2 - Computer Systems

S2 - Evaluation and Testing



LCCS Interwoven

S3 - ALT 2



S1 - Computers and Society

S2 - Data

S1 - Designing and Developing

S2 - Abstraction

S1 - Computational thinking

S2 - Evaluation and Testing



S1: Computers and society

Students learn about:

Social and ethical considerations of computing technologies

Machine learning
Artificial intelligence

User-centred design

The Internet

Turing machines



Students should be able to:

- 1.11 - discuss the complex relationship between computing technologies and society including issues of ethics
- 1.12 - compare the positive and negative impacts of computing on culture and society
- 1.13 - identify important computing developments that have taken place in the last 100 years and consider emerging trends that could shape future computing technologies
- 1.14 - explain when and what machine learning and AI algorithms might be used in certain contexts
- 1.15 - consider the quality of the user experience when interacting with computers and list the principles of universal design, including the role of a user interface and the factors that contribute to its usability
- 1.16 - compare two different user interfaces and identify different design decisions that shape the user experience**
- 1.17 - describe the role that adaptive technology can play in the lives of people with special needs
- 1.18 - recognise the diverse roles and careers that use computing technologies



NCCA - The evolution of computers in society

This booklet provides a chronologically structured series of detailed resources and learning plans aimed at supporting LCCS teachers as they explore the topic of Computers and Society with their students.





Stimulate a Debate strategy

We are now going to look at the *Stimulate a Debate* strategy as a pedagogical tool suited to the LCCS classroom.

We will:

1. define the strategy and the reasons for utilising it in your class
2. quickly walk through a sample *Stimulate a Debate* lesson
3. break into groups and examine specific *Stimulate a Debate* lessons from the *Evolution of Computers in Society* booklet
4. record our ideas, understandings, comments and adjustments in a shared file to which we will all have access



Stimulate a Debate strategy

A pedagogical framework through which complex ideas and concepts can be developed and understood by the class as a whole and each student separately.

Speaks to the constructivist ethos underpinning the delivery of LCCS.

Facilitation methodologies can be chosen to suit the particular needs of the students in your classroom.



Stimulate a Debate strategy

1. Watch a stimulus video/read a stimulus piece.
2. Provide prompt questions to provoke class discussion and elicit initial viewpoints.
3. Divide into research groups to explore the topic from key standpoints.
4. Choose a teaching/facilitation methodology.



Stimulate a Debate on how much data, and the kinds of data, governments and giant multi-nationals should be allowed to keep on citizens and consumers

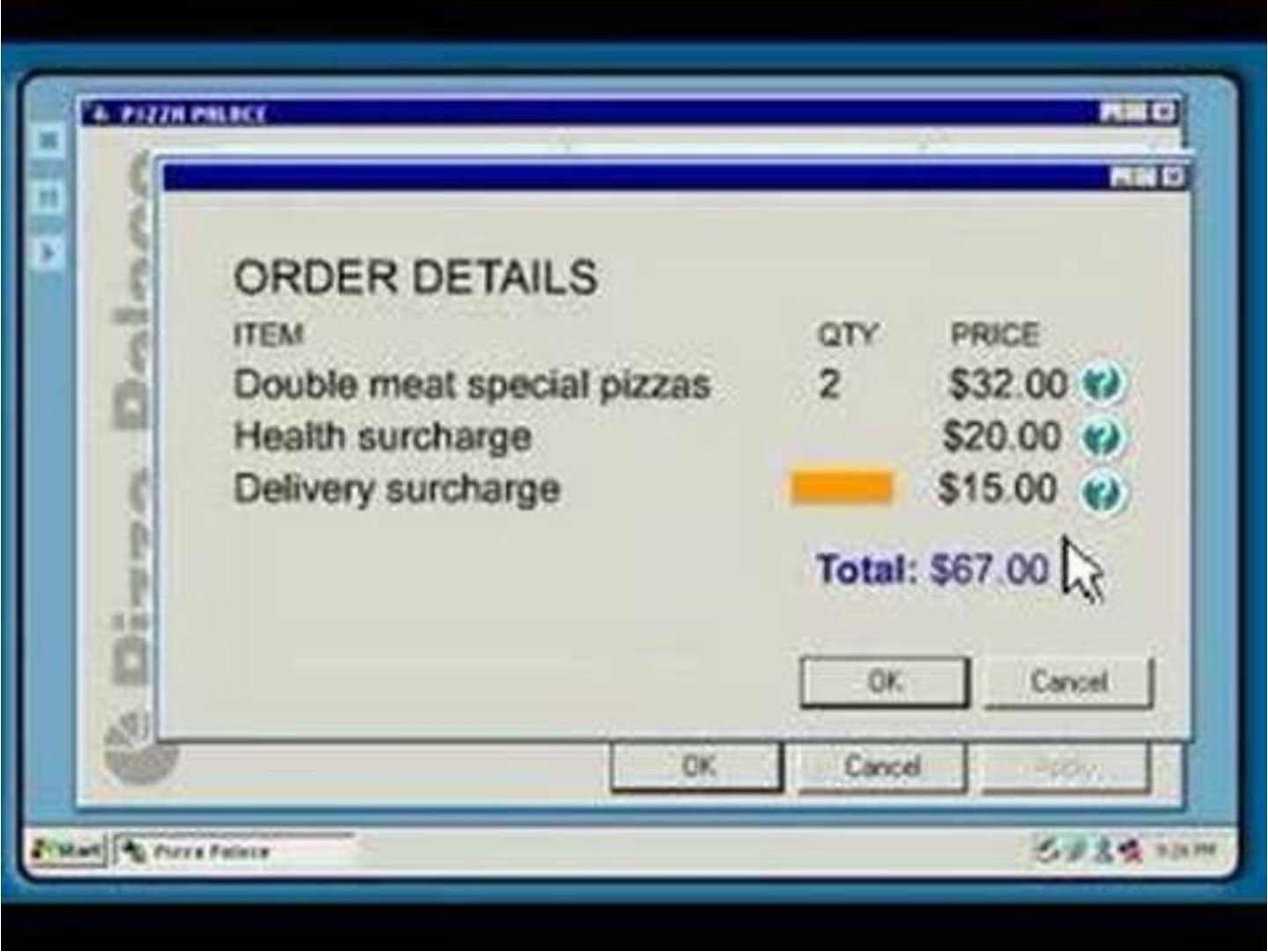
(NCCA booklet pgs. 51/52)



Step 1

Watch a stimulus video or read a stimulus piece.

Ordering a pizza





Step 2

Prompt questions to provoke class discussion and elicit initial viewpoints.



Sample prompt questions...

- What were the pros and cons of the pizza company being able to access the customer's data?
- Did they have too much access to data or was their usage of the data at fault?
- How much data do you think governments or high-tech companies **NEED** to retain on citizens? Perhaps a certain amount is required for national security? Or, in the case of online medical companies, could they use AI on your data to prevent future illness, mine your data or suggest treatments?



Step 3

Divide into research groups to explore the topic from key standpoints.



Key standpoints e.g. the ethics of data collection

- Governments should be allowed unlimited access to citizens' online data.
- Major institutions, such hi-tech medical or financial firms, should be allowed no access to consumer online data.



Step 4

Choose a Teaching/Facilitation methodology



Example facilitation technique: Jigsaw

Students first research each topic in research groups of three.

Use a Jigsaw Learning Technique to create groups of three comprising one student from three different themes. Each person discusses their research within their new group.

Reassemble into original groups and discuss any new learnings.



Recapping on the four steps:

1. Watch a stimulus video/read a stimulus piece.
2. Prompt questions to provoke class discussion and elicit initial viewpoints.
3. Divide into research groups to explore the topic from key standpoints.
4. Choose a teaching/facilitation methodology.



Main Group Activity

Stimulate a Debate – Lesson Discussion

Each group will work on a specific debate topic from the “*Evolution of Computers in Society*” booklet.

They will examine each step of the *Facilitate a Debate* strategy and discuss/comment/improve/amend as necessary.



Groups will record their work on a shared document, the results of which will be collated and shared via Slack in the coming days.



Main Group Activity

Stimulate a Debate – Lesson Discussion

- Some scaffold questions have been placed in the shared slides to help you.
- Please allow your own specific context and experience to guide your contributions and understanding, while creating this resource for your peers.
- Each group should nominate a *timekeeper*, *note-taker*, *spokesperson* and *discussion chair* in order to facilitate successful group work and feedback.





Main Group Activity

Stimulate a Debate – Lesson Discussion

You will have 30 minutes to complete the four steps of the strategy.

We will then reconvene to share thoughts and ideas on the process as a whole group.



30 Minutes



Group activity



Groups 1 & 2	The positive and negative impact of military innovations on both society and technological developments.	Page 19
Groups 3 & 4	How Computer Science is changing our world	Page 29
Groups 5 & 6	How should we respond as a society to the rapid growth of Artificial Intelligence	Page 34



10 Minutes



Oide



An Roinn Oideachais
Department of Education





Oide

Tacú leis an bhFoghlaim
Ghairmiúil i measc Ceannairí
Scoile agus Múinteoirí

Supporting the Professional
Learning of School Leaders
and Teachers

National Workshop 4

Session 2

Algorithms II





Part 1

Unconscious bias and algorithmic bias

Part 2





Introduction to algorithms (revisited)

Part 3

Searching algorithms

By the end of this session participants will have:



	participated in an interactive dialogue on unconscious bias and algorithmic bias
	revisited the definition and characteristics of algorithms, as well as the ubiquitous nature of algorithms in today's society
	participated in a coding activity relating to the linear search algorithm
	developed their understanding of the binary search algorithm



Key



Booklet



Individual Activity



Group Activity



Reflection



Group Feedback



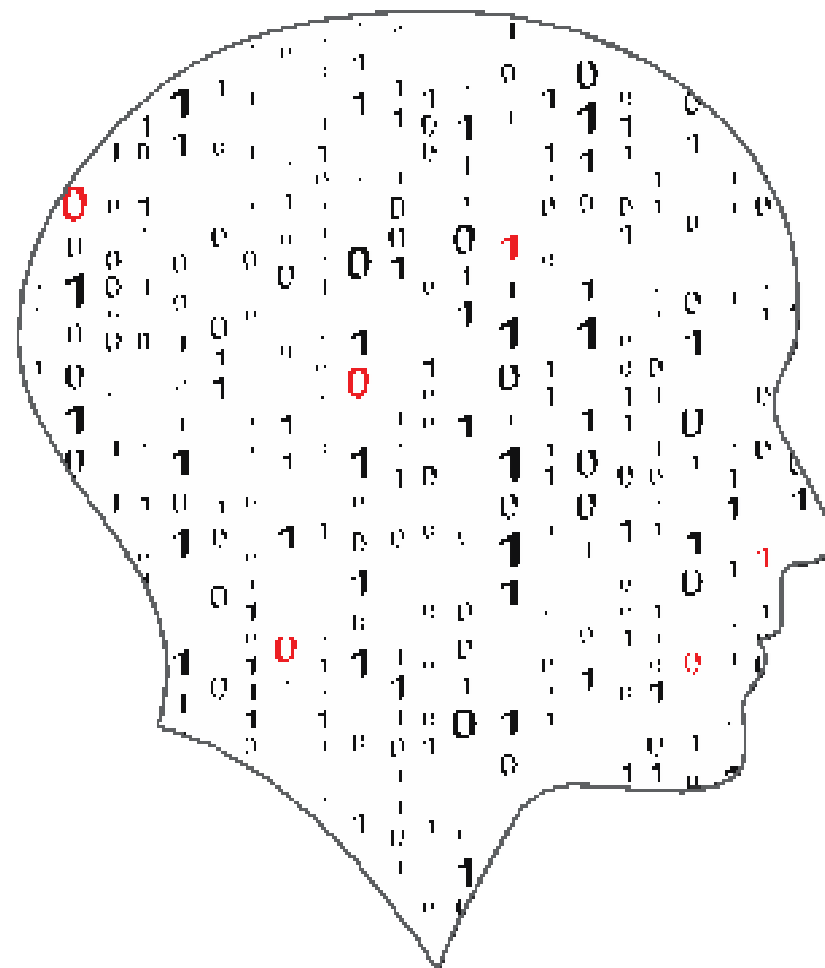
Oide

Tacú leis an bhFoghlaim
Ghairmiúil i measc Ceannairí
Scoile agus Múinteoirí

Supporting the Professional
Learning of School Leaders
and Teachers

Part 1

Unconscious Bias & Algorithmic Bias



LEAVING CERTIFICATE
COMPUTER SCIENCE



The Builder, the Nurse and the Doctor



A builder, leaning out of the van, shouts “nice legs” to a nurse passing by. The same nurse arrives at work, and casually mentions this to a senior doctor. The doctor said, “I’d never say that”.

The doctor has two grown up children who are 22 and 30. They get on very well. One is a Sergeant in the Army; the other is training to be a beauty therapist. The doctor divorced last year and is currently dating someone else.





	True	False	Don't know
The builder was driving a van			✓
The van was travelling quicker than the nurse			✓
There was at least one man in the van			✓
Not every man mentioned would shout "nice legs"			✓
The doctor is no longer living with his wife			✓
The doctor has a new girlfriend			✓
The doctor's son is in the army			✓
The youngest child is training to be a beauty therapist			✓
At some point a man spoke to a woman			✓
At least two of the people mentioned are men			✓
A woman was shouted at			✓



What is unconscious bias?

- Natural
- Rapid categorization of people
- Created by social influence
- Unintentional
- Used by everyone
- Most likely to be acted on when we are stressed or tired
- A bad thinking habit

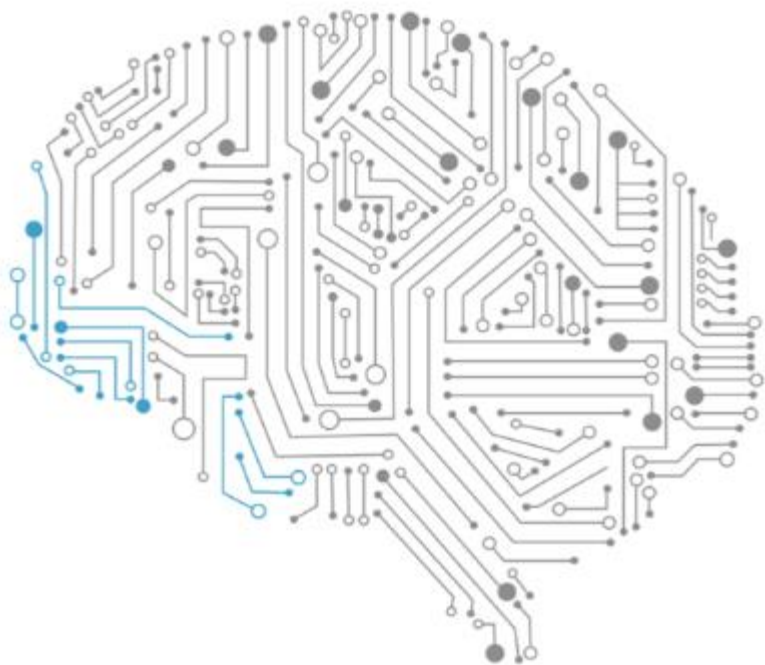




What does unconscious bias have to do with algorithms?

*“Some of the biggest problems in the industry aren’t technical – they’re people (egos etc.) diversity creates better solutions and opportunities because of wider experience set, perspectives and **less bias ...**”*

James Whelton, CoderDojo Co-Founder





Algorithmic bias

“The nurse said that...”

“The pilot said that...”

“The computer programmer said that...”





The Feedback Loop



“You are not only allowing it (the computer system) to make a decision, you are also allowing it to determine/create the future they have predicted”



What does unconscious bias have to do with YOU?





Personal reflection



What are my values as a teacher?

“We do not learn from experience... we learn from reflecting on experience.” (John Dewey)

“Sincerity of intention does not guarantee purity of practice.”
(Stephen Brookfield)



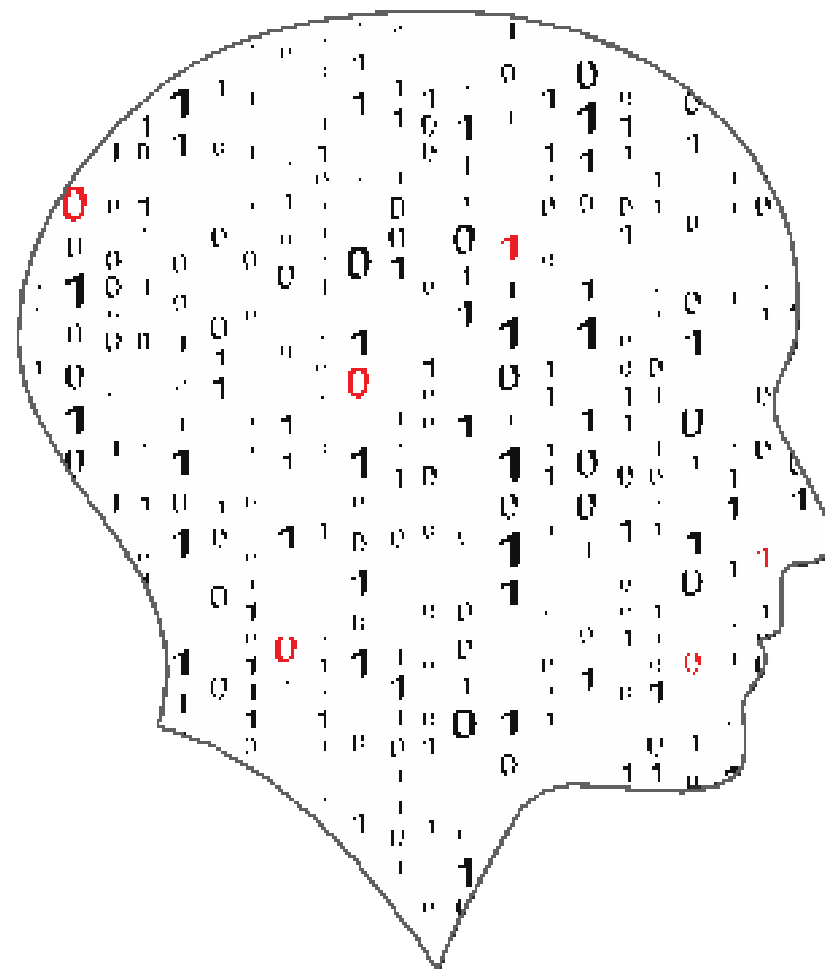
Oide

Tacú leis an bhFoghlaim
Ghairmiúil i measc Ceannairí
Scoile agus Múinteoirí

Supporting the Professional
Learning of School Leaders
and Teachers

Part 2

Introduction to Algorithms - revisited



LEAVING CERTIFICATE
COMPUTER SCIENCE





Algorithms and the Specification

“Computer science is the study of computers and algorithmic processes. Leaving Certificate Computer Science includes how programming and computational thinking can be applied to the solution of problems, and how computing technology impacts the world around us. “

Strand 1: Practices and principles	Strand 2: Core concepts	Strand 3: Computer science in practice
<ul style="list-style-type: none">▶ Computers and society▶ Computational thinking▶ Design and development	<ul style="list-style-type: none">▶ Abstraction▶ Algorithms▶ Computer systems▶ Data▶ Evaluation/Testing	<ul style="list-style-type: none">▶ Applied learning task 1<ul style="list-style-type: none">- Interactive information systems▶ Applied learning task 2 - Analytics▶ Applied learning task 3<ul style="list-style-type: none">- Modelling and simulation▶ Applied learning task 4<ul style="list-style-type: none">- Embedded systems

NCCA Curriculum specification, Page 1

LCCS Learning Outcomes



Oide

2.5 use pseudo code to outline the functionality of an algorithm

2.6 construct algorithms using appropriate sequences, selections/conditionals, loops and operators to solve a range of problems, to fulfil a specific requirement

2.7 implement algorithms using a programming language to solve a range of problems

2.8 apply basic search and sorting algorithms and describe the limitations and advantages of each algorithm

2.9 assemble existing algorithms or create new ones that use functions (**including recursive**), procedures, and modules

2.10 explain the common measures of algorithmic efficiency using any algorithms studied

S2: Algorithms

Programming concepts

Sorting: Simple sort, Insert sort, Bubble sort, **Quicksort**

Search: Linear search, Binary search

Algorithmic complexity

See also learning outcomes 1.6, 1.7 1.14, 1.22, 2.3, 3.4 and 3.7 ... plus others



What is an Algorithm?

A step-by-step procedure for solving a problem or accomplishing some end especially by a computer.

Merriam-Webster

- ✓ A sequence of instructions
- ✓ A way of capturing intelligence and sharing it with others
- ✓ Provide general solutions to problems
- ✓ Some problems are so hard that they cannot be solved by algorithms (computability)
- ✓ Can be expressed in a variety of different ways
- ✓ Common elements include input, processing, output
- ✓ Close relationship between algorithms and data structures
- ✓ Essential features are correctness and effectiveness
- ✓ Rule-based algorithms vs. Machine learning algorithms (AI)



What is an algorithm?

“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 a machine”

Source: Knuth, D The Art of Computer Programming (Vol. 1, Fundamental Algorithms, 3rd ed.)

According to Knuth, an algorithm has five important features:

Finiteness

An algorithm must always terminate after a finite number of steps.

Definiteness

Each step must be precisely defined.

Input

An algorithm has zero or more inputs.

Output

An algorithm has one or more outputs, which have a specified relation to the inputs.

Effectiveness

All operations to be performed must be sufficiently basic that they can in principle be done exactly and in finite length of time by someone using pencil and paper.



Donald Knuth



Activity: Features of Algorithms

“What features of an algorithm are demonstrated in the video?”

In what other contexts do you think the Gale-Shapley algorithm could be used?





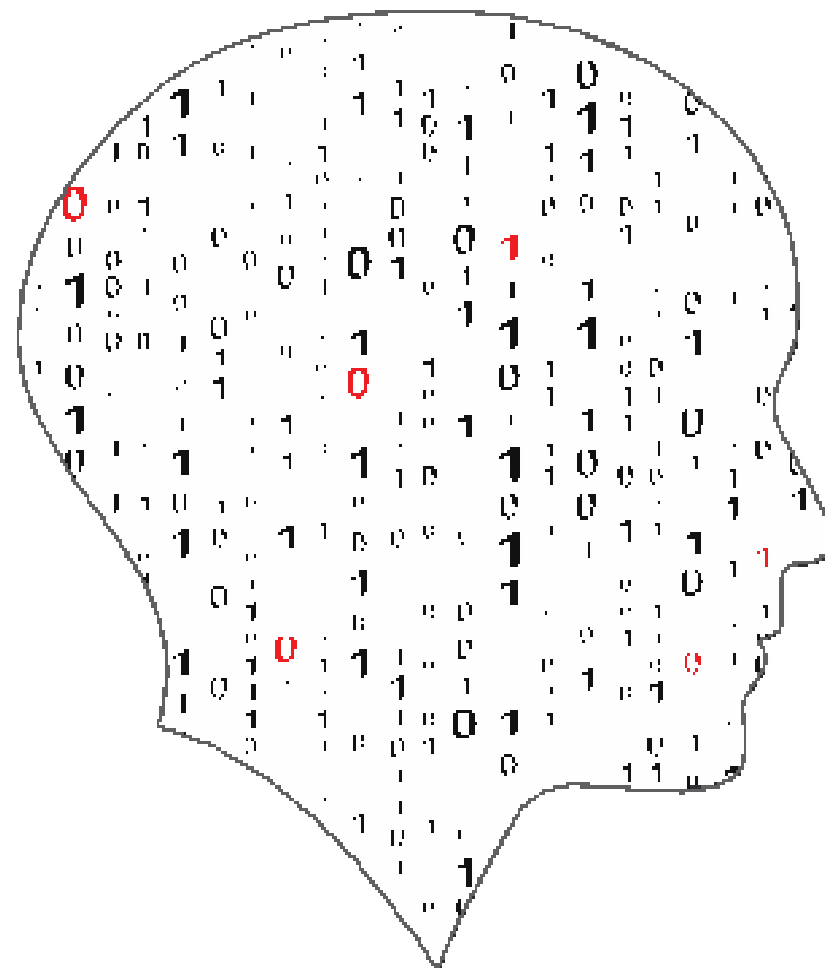
Oide

Tacú leis an bhFoghlaim
Ghairmiúil i measc Ceannairí
Scoile agus Múinteoirí

Supporting the Professional
Learning of School Leaders
and Teachers

Part 3

Searching Algorithms: Linear search



LEAVING CERTIFICATE
COMPUTER SCIENCE



LCCS Learning Outcomes



Oide

2.5 use pseudo code to outline the functionality of an algorithm

2.6 construct algorithms using appropriate sequences, selections/conditionals, loops and operators to solve a range of problems, to fulfil a specific requirement

2.7 implement algorithms using a programming language to solve a range of problems

2.8 apply basic search and sorting algorithms and describe the limitations and advantages of each algorithm

2.9 assemble existing algorithms or create new ones that use functions (including recursive), procedures, and modules

2.10 explain the common measures of algorithmic efficiency using any algorithms studied

S2: Algorithms

Programming concepts

Sorting: Simple sort, Insert sort, Bubble sort, **Quicksort**

Search: Linear search, Binary search

Algorithmic complexity

See also learning outcomes 1.6, 1.7 1.14, 1.22, 2.3, 3.4 and 3.7 ... plus others



Searching

An algorithm that maps the following input/output pair is called a search algorithm:

Input: An array, A , that contains n orderable elements (often called *keys*) $A[0, 1, \dots, n-1]$ and some target value commonly referred to as an *argument*.

Output: If the argument is found in A it is conventional to return its zero-based positional offset (i.e. the index) and if the argument is not found some implementations return the length of the list while others return -1 . (Either of these two outputs can be used to indicate that the argument doesn't exist in A .)

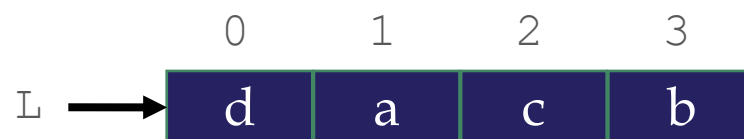


Searching

For example, a search to find argument 'c' in the list L below would return 2.



A search to find argument 'z' (or any other value not on L) in L would return either -1 or 4 (the length of the list).



List traversal



Pass over each element in the list one at a time

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

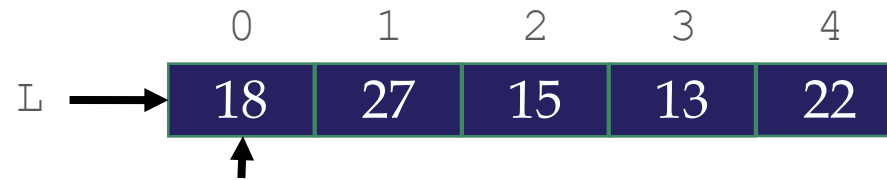
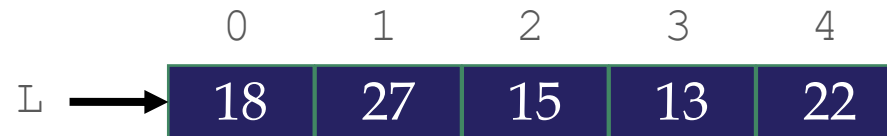
```
for index in range(len(L)):  
    print(L[index])
```

Equivalent to ...

```
for v in L:  
    print(v)
```

```
index = 0  
while index < len(L):  
    print(L[index])  
    index = index + 1
```

```
print(L[0])  
print(L[1])  
print(L[2])  
print(L[3])  
print(L[4])
```



Output Displayed

18

27

15

13

22



Linear Search

Question: does the list below contain the number 14?

15	4	41	13	24	14	12	21
----	---	----	----	----	----	----	----

If **14 is found**, what should the result be?

True?

The position (index) of 14?

If **14 is not found**, what should the result be?

False?

-1?

The length of the list?



Input:

A list L and a *target value* of 14

	0	1	2	3	4	5	6	7
L →	15	4	41	13	24	14	12	21



Input:

A list L and a *target value* of 14

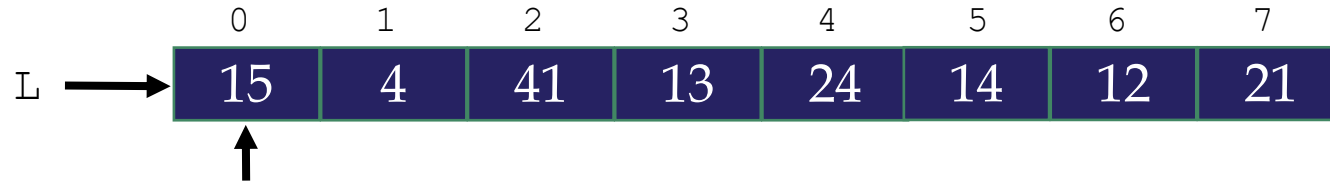
	0	1	2	3	4	5	6	7
L →	15	4	41	13	24	14	12	21

Required Output:

If the target value is found in L, its *index*

If the target value is not found in L, -1 is returned

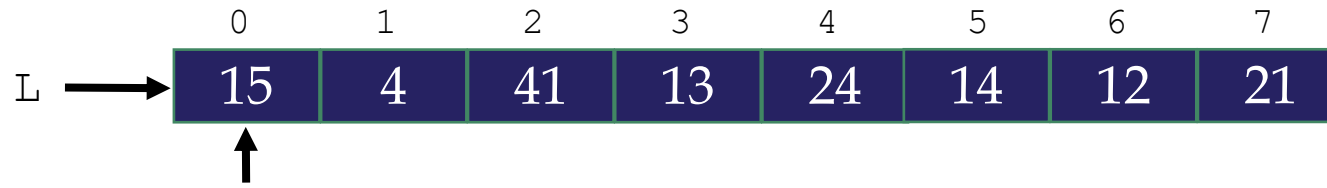
Start at the first element and ask is $L[0] ==$ the *target value*?



Start at the first element and ask is $L[0] ==$ the *target value*?

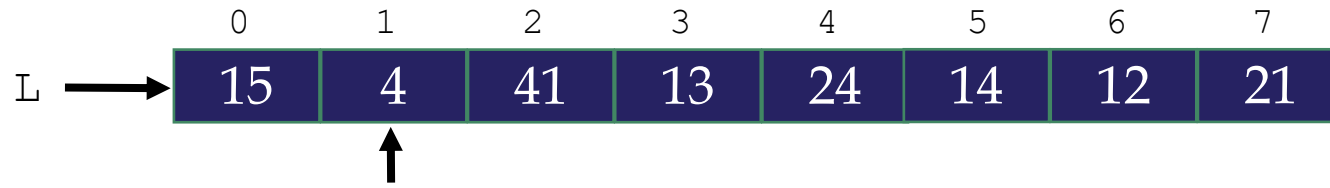


Oide



$L[0] == 14?$ NO!

Move to the next element and ask is $L[1] ==$ the *target value*?

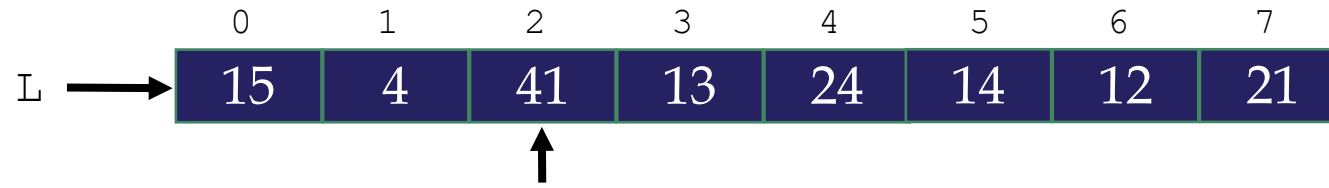


$L[1] == 14?$ NO!

Move to the next element and ask is $L[2] ==$ the *target value*?

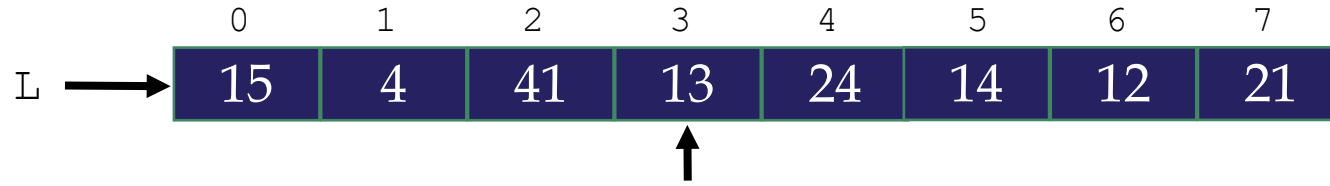


Oide



$L[2] == 14?$ NO!

Move to the next element and ask is $L[3] ==$ the *target value*?

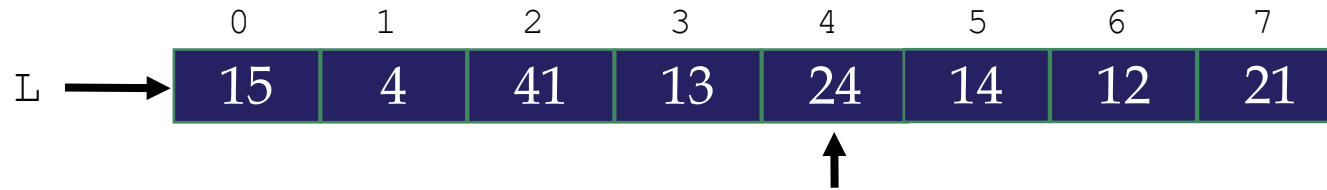


$L[3] == 14?$ NO!

Move to the next element and ask is $L[4] ==$ the *target value*?

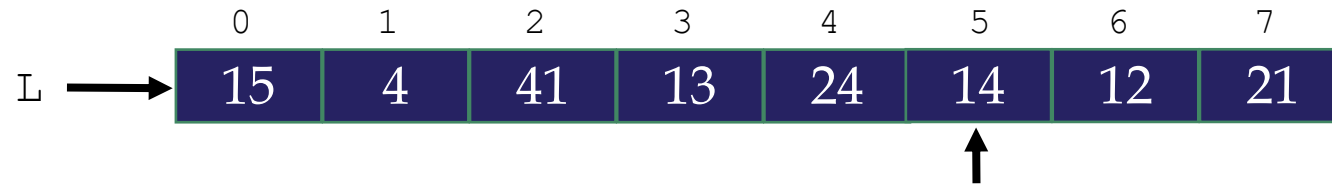


Oide

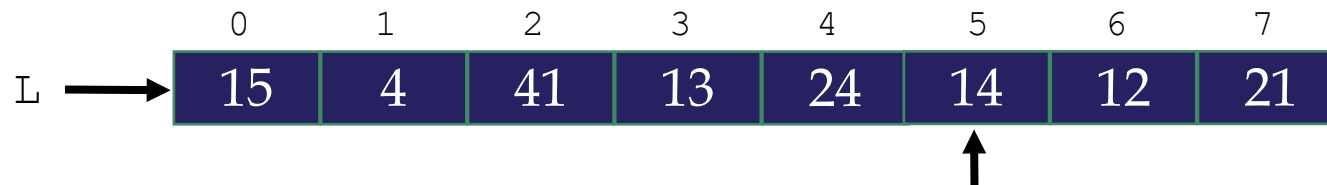


$L[4] == 14?$ NO!

Move to the next element and ask is $L[5] ==$ the *target value*?



$L[5] == 14?$ YES!



L[5] == 14? YES!

We have found the *target value* at index 5

The result of the search is 5



Breakout task



PRIMM activity on linear search

```
1. # A program to perform a linear search
2.
3. # linearSearch function
4. # takes two parameters: num is the target value, myList is the list to search
5. # if target value found, returns position (index) of 'num'; otherwise, returns -1
6. def linearSearch(num, myList):
7.     for i in range(len(myList)):
8.         if myList[i] == num:
9.             return i
10.    return -1
11.
12. # PYTHON STARTS EXECUTING FROM HERE ...
13. listToSearch = [1,2,3,4,5,6,7,8,9,10]
14. userNum = int(input("Enter the number you want to find:"))
15.
16. # function call
17. position = linearSearch(userNum, listToSearch)
18.
19. # display the result of the linear search
20. print("The number is at position", position)
```



20 minutes



P 9



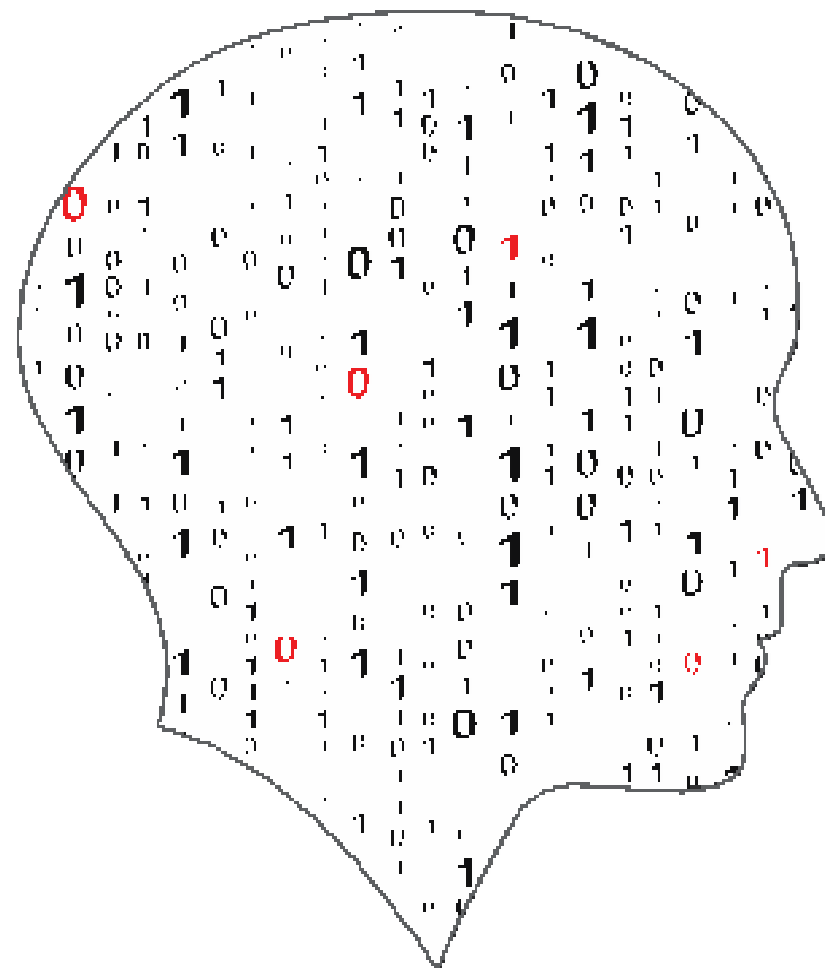
Oide

Tacú leis an bhFoghlaim
Ghairmiúil i measc Ceannairí
Scoile agus Múinteoirí

Supporting the Professional
Learning of School Leaders
and Teachers

Searching Algorithms

Binary search



LEAVING CERTIFICATE
COMPUTER SCIENCE





Input:

A list L and a target value of 28

	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
L →	2	4	5	7	8	9	12	14	17	19	22	25	27	28	33	37



Input:

A list L and a *target value* of 28

	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
L →	2	4	5	7	8	9	12	14	17	19	22	25	27	28	33	37

Required Output:

If the *target value* is found in L, its index

If the *target value* is not found in L, we return -1

Binary Search Algorithm



Oide

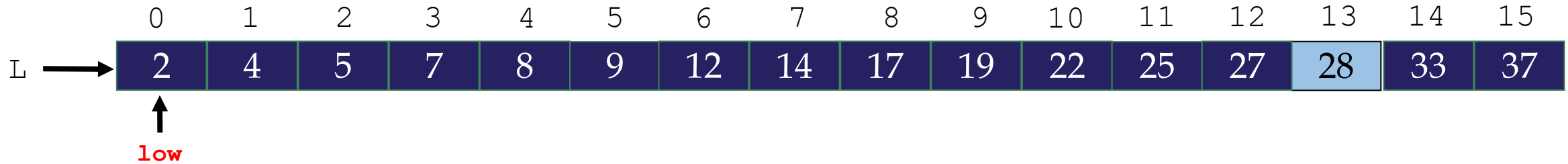
	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
L →	2	4	5	7	8	9	12	14	17	19	22	25	27	28	33	37

Pseudo-code: (target value is 28)

Binary Search Algorithm



Oide



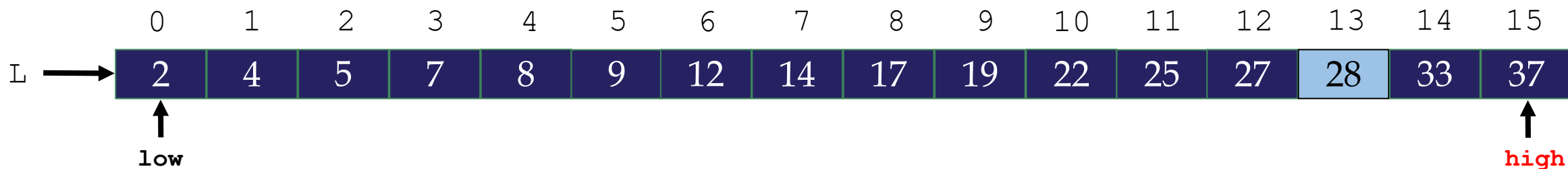
Pseudo-code: (target value is 28)

1. Set low = 0

Binary Search Algorithm



Oide



Pseudo-code: (target value is 28)

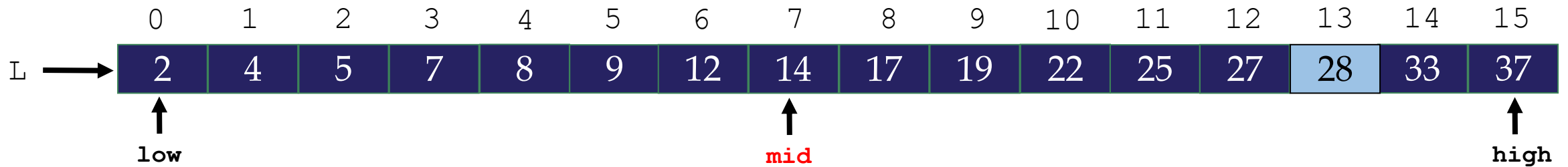
1. Set low = 0

2. Set high = length of list - 1

Binary Search Algorithm



Oide



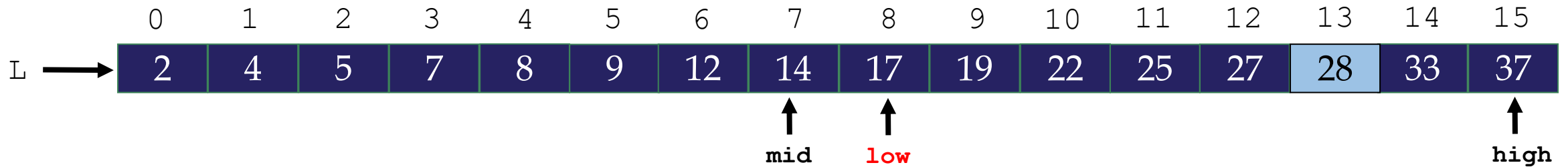
Pseudo-code: (target value is 28)

1. Set `low = 0`
2. Set `high = length of list - 1`
3. Set `mid = $\frac{low+high}{2}$, rounded down to an integer`

Binary Search Algorithm



Oide



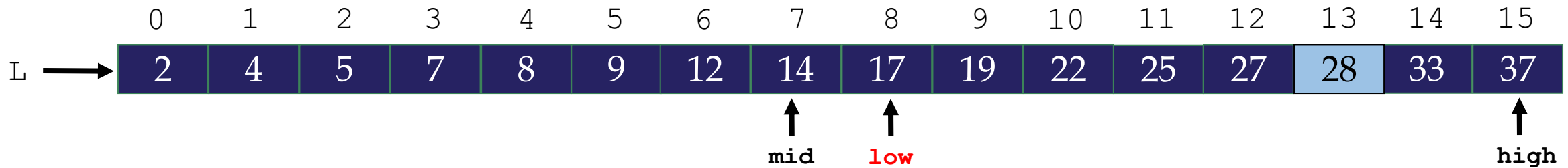
Pseudo-code: (target value is 28)

1. Set low = 0
2. Set high = length of list - 1
3. Set $mid = \frac{low+high}{2}$, rounded down to an integer
4. If the value at the mid position is the same as the target value
Return mid
Else If the value at the mid position is less than the target value
Set low = mid + 1
Else If the value at the mid position is greater than the target value
Set high = mid - 1

Binary Search Algorithm



Oide



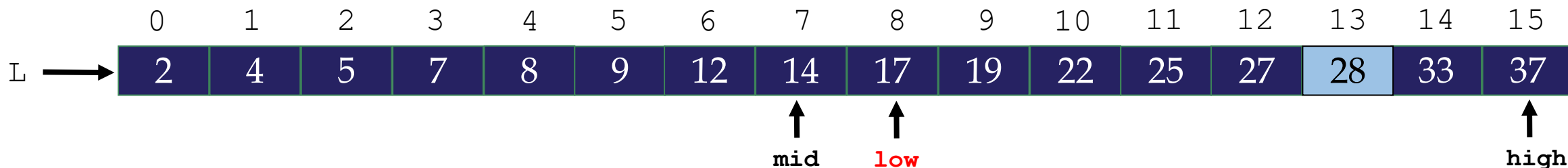
Pseudo-code: (target value is 28)

1. Set low = 0
2. Set high = length of list - 1
3. Set $mid = \frac{low+high}{2}$, rounded down to an integer
4. If the value at the mid position is the same as the target value
Return mid
Else If the value at the mid position is less than the target value
Set low = mid + 1
Else If the value at the mid position is greater than the target value
Set high = mid - 1
5. As long as low doesn't 'cross over' high, go back to step 3 above

Binary Search Algorithm



Oide



Pseudo-code: (target value is 28)

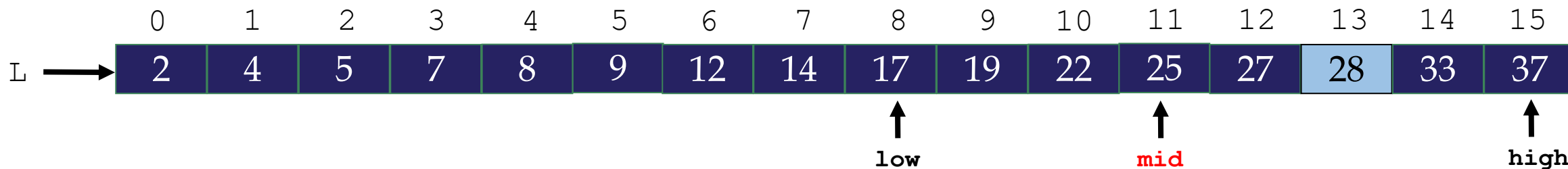
1. Set low = 0
2. Set high = length of list - 1
3. Set $mid = \frac{low+high}{2}$, rounded down to an integer
4. If the value at the mid position is the same as the target value
Return mid
Else If the value at the mid position is less than the target value
Set low = mid + 1
Else If the value at the mid position is greater than the target value
Set high = mid - 1
5. As long as low doesn't 'cross over' high, go back to step 3 above

↓
In Python this means, while low <= high:

Binary Search Algorithm



Oide



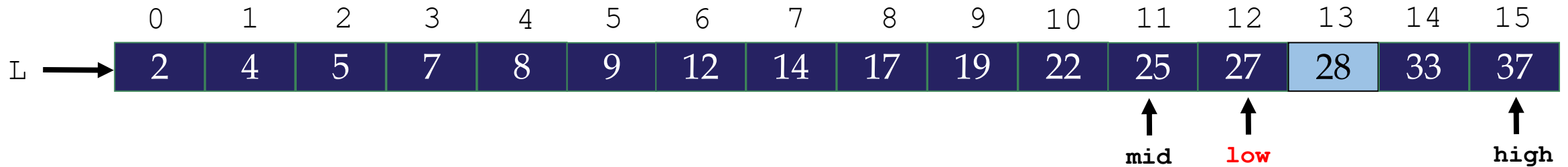
Pseudo-code: (target value is 28)

1. Set low = 0
2. Set high = length of list - 1
3. Set $mid = \frac{low+high}{2}$, rounded down to an integer
4. If the value at the mid position is the same as the target value
Return mid
Else If the value at the mid position is less than the target value
Set low = mid + 1
Else If the value at the mid position is greater than the target value
Set high = mid - 1
5. As long as low doesn't 'cross over' high, go back to step 3 above

Binary Search Algorithm



Oide



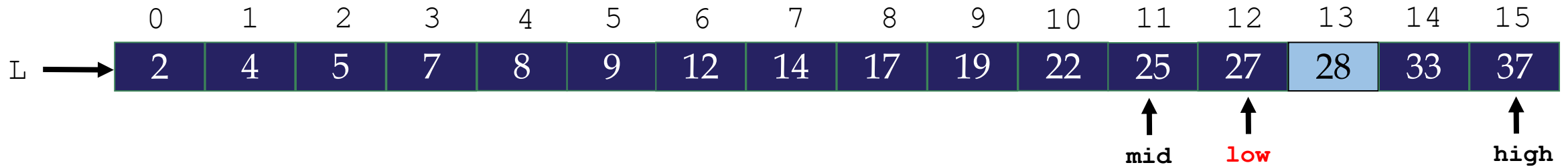
Pseudo-code: (target value is 28)

1. Set low = 0
2. Set high = length of list - 1
3. Set $mid = \frac{low+high}{2}$, rounded down to an integer
4. If the value at the mid position is the same as the target value
Return mid
Else If the value at the mid position is less than the target value
Set low = mid + 1
Else If the value at the mid position is greater than the target value
Set high = mid - 1
5. As long as low doesn't 'cross over' high, go back to step 3 above

Binary Search Algorithm



Oide



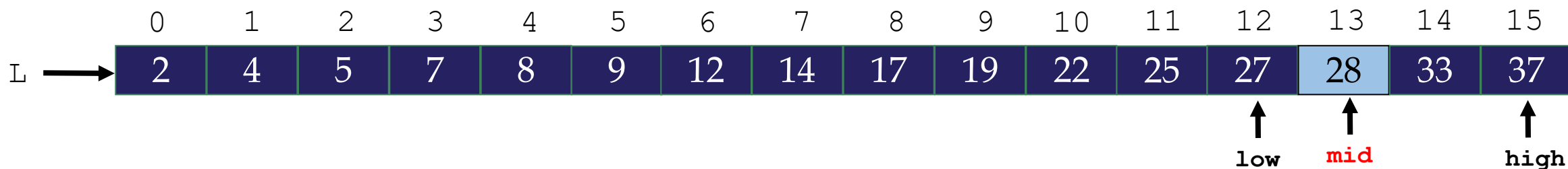
Pseudo-code: (target value is 28)

1. Set low = 0
2. Set high = length of list - 1
3. Set $mid = \frac{low+high}{2}$, rounded down to an integer
4. If the value at the mid position is the same as the target value
Return mid
Else If the value at the mid position is less than the target value
Set low = mid + 1
Else If the value at the mid position is greater than the target value
Set high = mid - 1
5. As long as low doesn't 'cross over' high, go back to step 3 above

Binary Search Algorithm



Oide



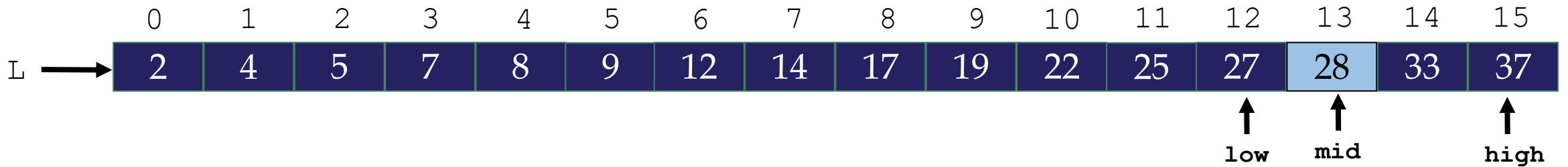
Pseudo-code: (target value is 28)

1. Set low = 0
2. Set high = length of list - 1
3. Set $mid = \frac{low+high}{2}$, rounded down to an integer
4. If the value at the mid position is the same as the target value
Return mid
Else If the value at the mid position is less than the target value
Set low = mid + 1
Else If the value at the mid position is greater than the target value
Set high = mid - 1
5. As long as low doesn't 'cross over' high, go back to step 3 above

Binary Search Algorithm



Oide



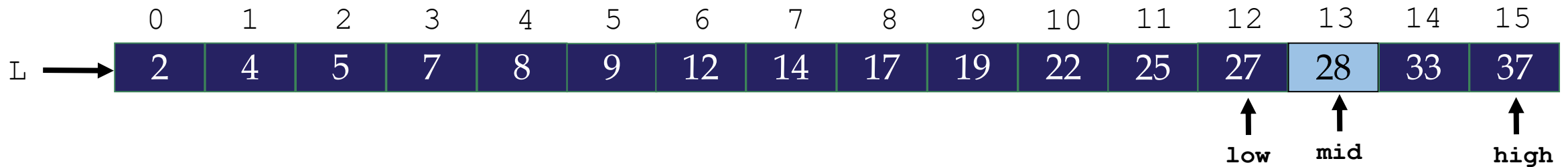
Pseudo-code: (target value is 28)

1. Set low = 0
2. Set high = length of list - 1
3. Set $mid = \frac{low+high}{2}$, rounded down to an integer
4. If **the value at the mid position is the same as the target value**
Return mid
Else If the value at the mid position is less than the target value
Set low = mid + 1
Else If the value at the mid position is greater than the target value
Set high = mid - 1
5. As long as low doesn't 'cross over' high, go back to step 3 above

Binary Search Algorithm



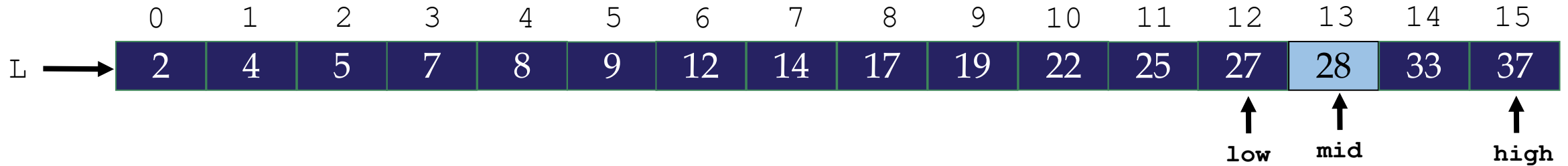
Oide



Pseudo-code: (target value is 28)

1. Set low = 0
2. Set high = length of list - 1
3. Set $mid = \frac{low+high}{2}$, rounded down to an integer
4. If the value at the mid position is the same as the target value
Return mid
Else If the value at the mid position is less than the target value
Set low = mid + 1
Else If the value at the mid position is greater than the target value
Set high = mid - 1
5. As long as low doesn't 'cross over' high, go back to step 3 above

Binary Search Algorithm



Pseudo-code: (target value is 28)

1. Set low = 0
2. Set high = length of list - 1
3. Set $mid = \frac{low+high}{2}$, rounded down to an integer
4. If the value at the mid position is the same as the target value
Return mid
Else If the value at the mid position is less than the target value
Set low = mid + 1
Else If the value at the mid position is greater than the target value
Set high = mid - 1
5. As long as low doesn't 'cross over' high, go back to step 3 above

13 is returned (as it is the value of mid)
This is the index of the target element.

Q. How many comparisons were needed?

Q. How many comparisons would be needed for the linear search?



Find 'J'

0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23
A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X





An Roinn Oideachais
Department of Education





Oide

Tacú leis an bhFoghlaim
Ghairmiúil i measc Ceannairí
Scoile agus Múinteoirí

Supporting the Professional
Learning of School Leaders
and Teachers

National Workshop 4

Session 3

Computer Systems 3





Key



Booklet



Individual Activity



Group Activity



Reflection



Group Feedback

Session Overview



Section 1	Basic Electronics
Section 2	Logic Gates
Section 3	Group Activity



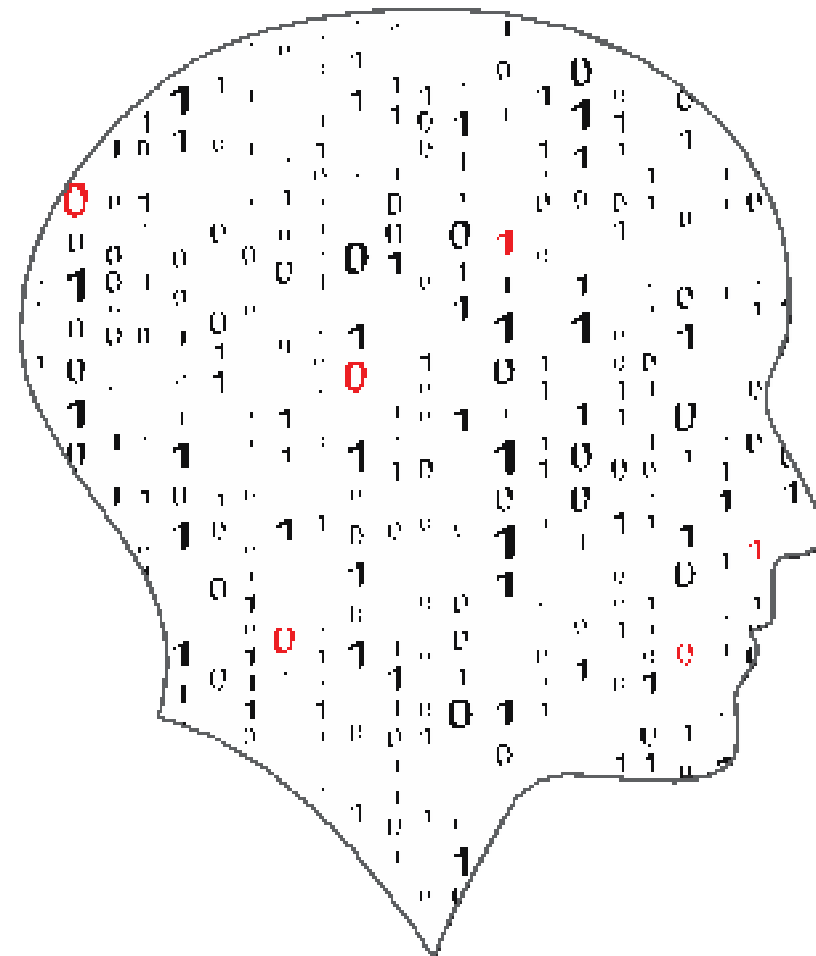
Oide

Tacú leis an bhFoghlaim
Ghairmiúil i measc Ceannairí
Scoile agus Múinteoirí

Supporting the Professional
Learning of School Leaders
and Teachers

Section I

Basic electronics



LEAVING CERTIFICATE
COMPUTER SCIENCE





By the end of this session participants will have:

- gained a deeper understanding of computer systems concepts and terminology including basic electronics, logic gates and circuits, truth tables, and computer components
- taken part in and reflected upon a half-adder activity
- acquired additional knowledge and ideas on how they will facilitate the learning of computer systems in their own classrooms



Context

Strand 1: Practices and principles	Strand 2: Core concepts	Strand 3: Computer science in practice
<ul style="list-style-type: none">▶ Computers and society▶ Computational thinking▶ Design and development	<ul style="list-style-type: none">▶ Abstraction▶ Algorithms▶ Computer systems▶ Data▶ Evaluation/Testing	<ul style="list-style-type: none">▶ Applied learning task 1<ul style="list-style-type: none">- Interactive information systems▶ Applied learning task 2 - Analytics▶ Applied learning task 3<ul style="list-style-type: none">- Modelling and simulation▶ Applied learning task 4<ul style="list-style-type: none">- Embedded systems

“The core concepts are developed theoretically and applied practically. In this way, conceptual classroom-based learning is intertwined with experimental computer lab-based learning throughout the two years of the course.”

NCCA Curriculum specification, Page 20

LCCS Learning Outcomes



S2: Computer systems

CPU: ALU, Registers, Program counter, Memory

Basic electronics: voltage, current, resistors, capacitors, transistors

2.11 describe the different components within a computer and the function of those components

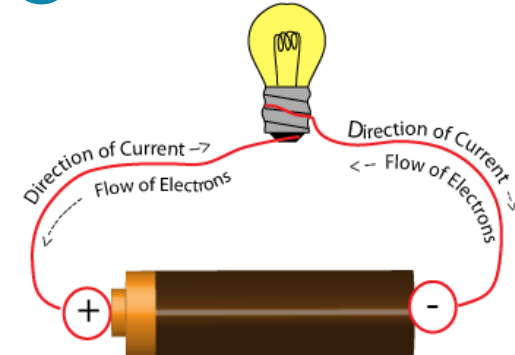
2.12 describe the different types of logic gates **and explain how they can be arranged into larger units to perform more complex tasks**

Students learn about:	Students should be able to:
Embedded systems	3.11 use and control digital inputs and outputs within an embedded system
Computing inputs and outputs	3.12 measure and store data returned from an analogue input
Computer systems	3.13 develop a program that utilises digital and analogue inputs
Design process	3.14 design automated applications using embedded systems



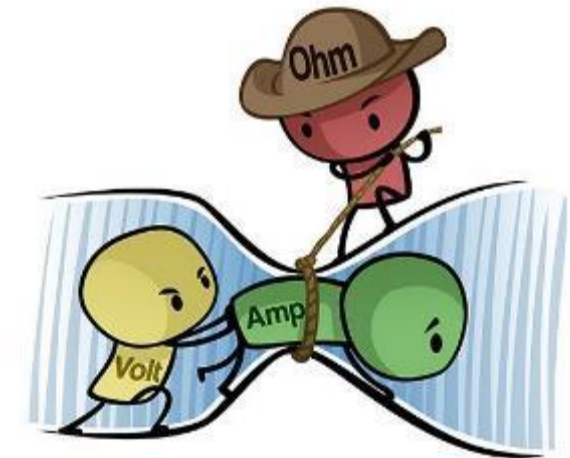
Current, Voltage and Resistance

Current is the flow of electrons (measured in Amperes or A)



Voltage describes the “pressure” that pushes electrons (measured in Volts or V)

Resistance is something that resists the flow of electrons (measured in Ohms or Ω)



Demonstrating Current, Voltage and Resistance



Oide

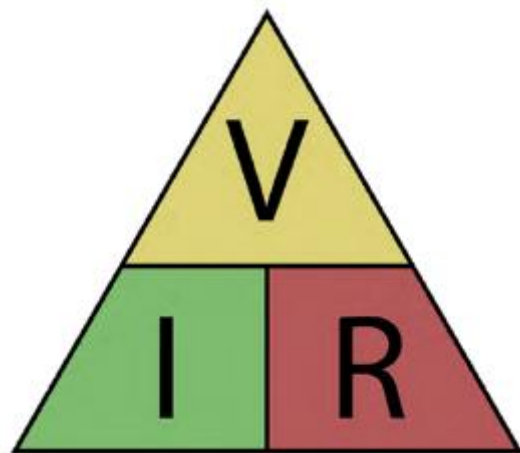
The screenshot displays a virtual circuit lab interface. On the left, a vertical toolbar contains icons for various components: Wire, Battery, AC Voltage, Light Bulb, Resistor, Capacitor, Inductor, and Switch. Below the toolbar are icons for a battery symbol, a plus/minus sign, a magnifying glass, and a zoom-in icon. The main workspace shows a rectangular circuit loop. A light bulb is connected to the bottom wire, emitting yellow light rays. The circuit includes a battery at the top, a resistor on the left, and a switch on the right. Blue dots representing electrons are shown moving through the wires. A text prompt at the bottom center reads "Tap circuit element to edit." On the right side, a control panel includes a "Show Current" section with a checked box, a radio button for "Electrons" (selected), and a radio button for "Conventional" with a red arrow. Below this are checkboxes for "Labels", "Values", and "Stopwatch". Further down are icons for a Voltmeter, Ammeter, Voltage Chart, and Current Chart. At the bottom right of the control panel is a green plus icon labeled "Advanced". At the very bottom of the interface, there is a speaker icon and the logo for "PhET".

Circuit Construction Kit: AC - Virtual Lab



What is the relationship between Voltage, Current and Resistance?

Demonstrate Ohm's Law



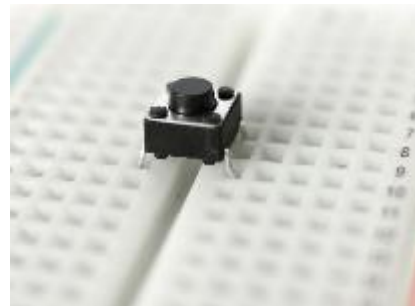
$V = I R$

Voltage: 4.5 V
Resistance: 500 Ω

current = 9.0 mA



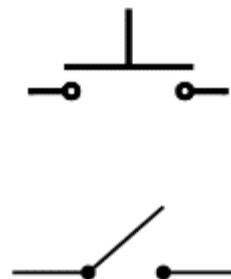
Electrical components



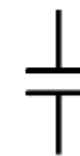
Light Emitting Diode (LED)



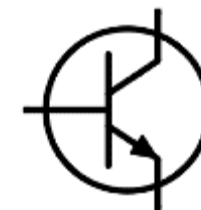
Resistor



Switch



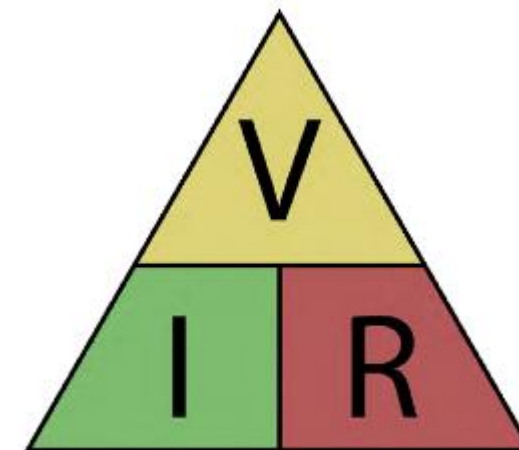
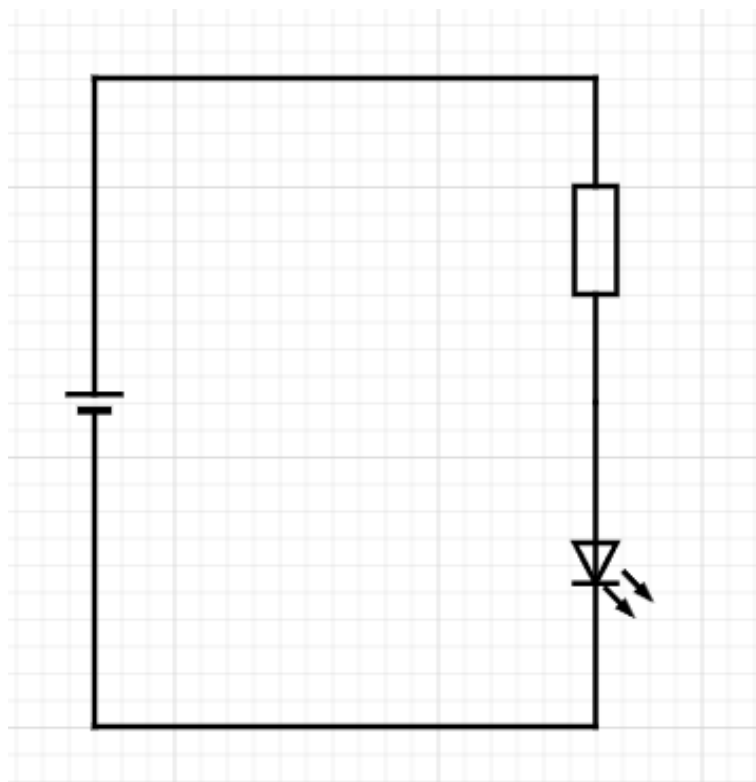
Capacitor



Transistor



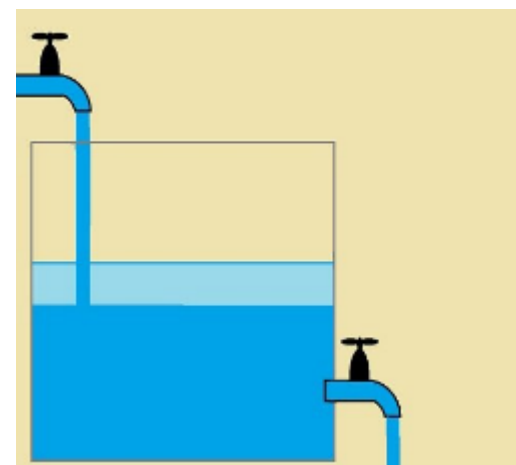
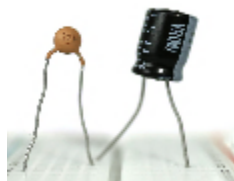
Circuit schematic



$$V = IR$$



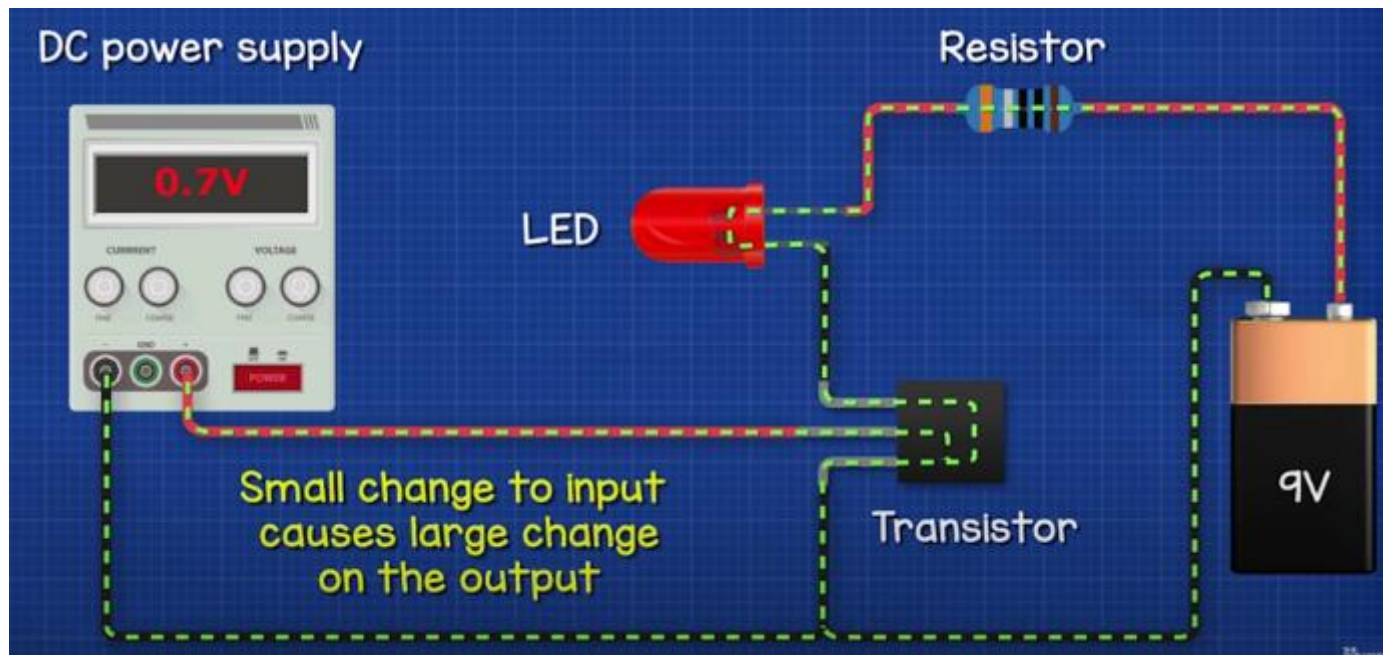
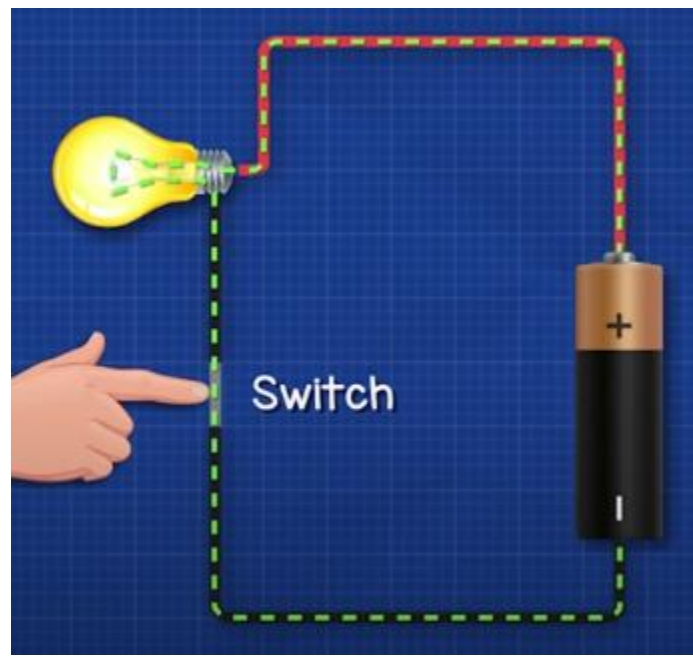
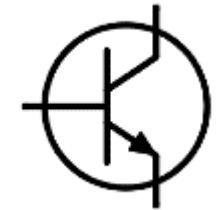
Capacitors



<https://www.youtube.com/watch?v=X4EUwTwZ110>



Transistors





https://youtu.be/J4oO7PT_nzQ



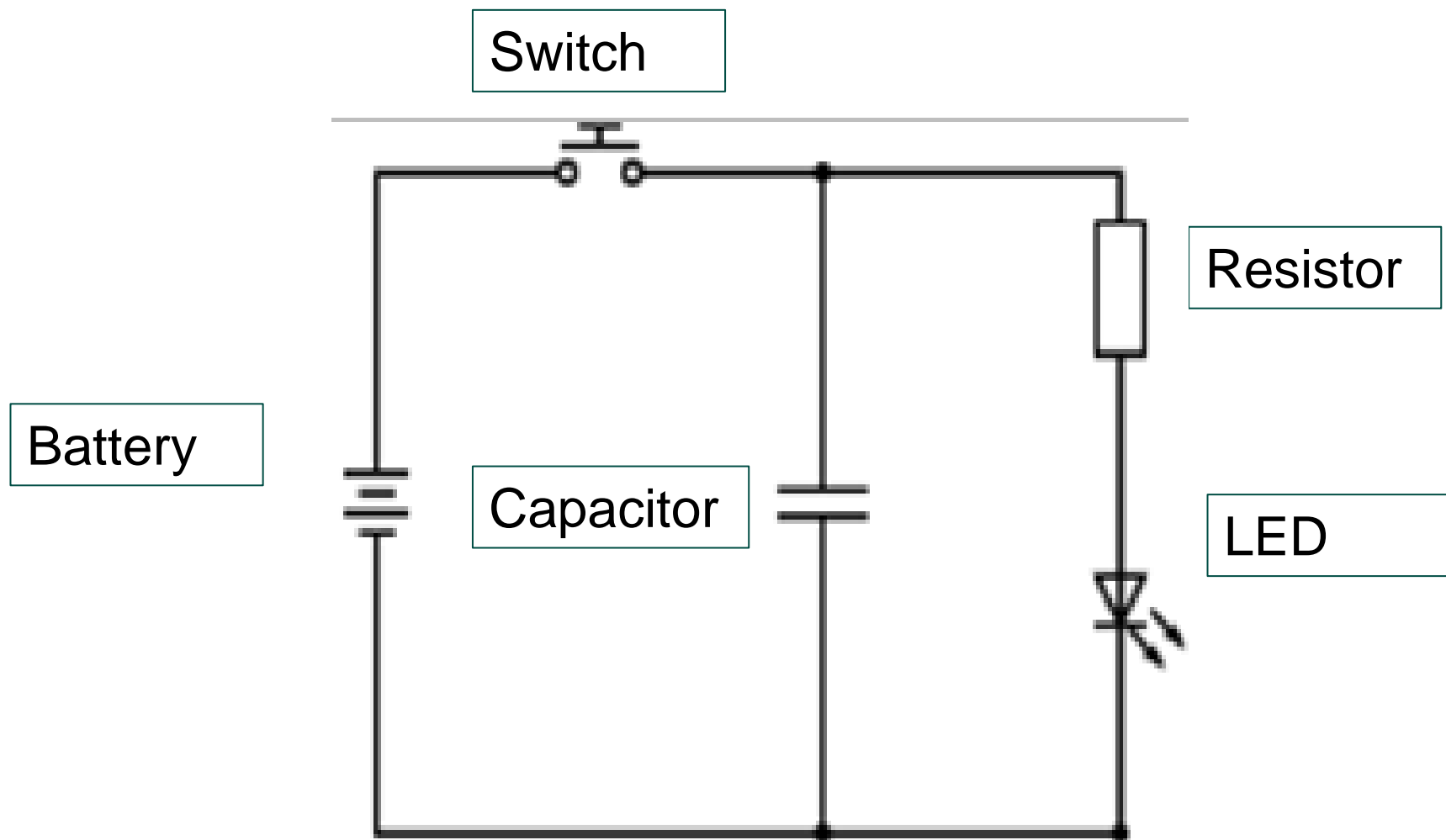
Basic Electronics



P. 16 & 17



Components in a circuit





Activity - Think Pair Share



Think

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



Pair

Participants turn to the person beside them to discuss their ideas

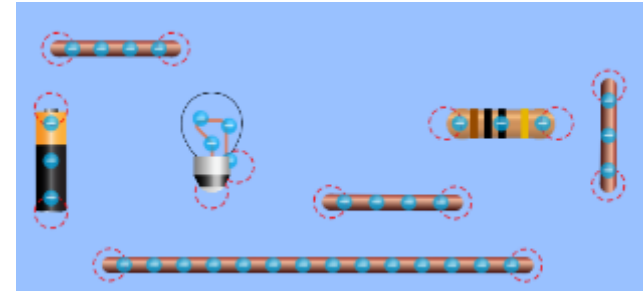


Share

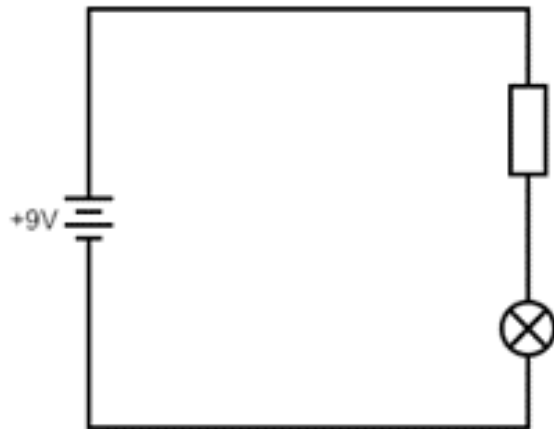
Pairs share their answers with other pairs (square) or the wider group



Circuit with Resistor

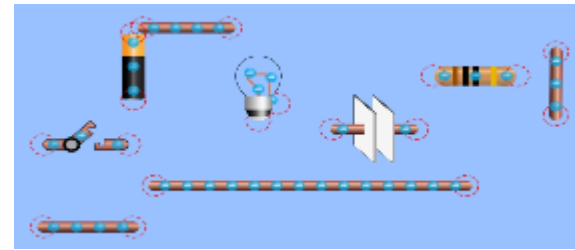


Phet Simulator

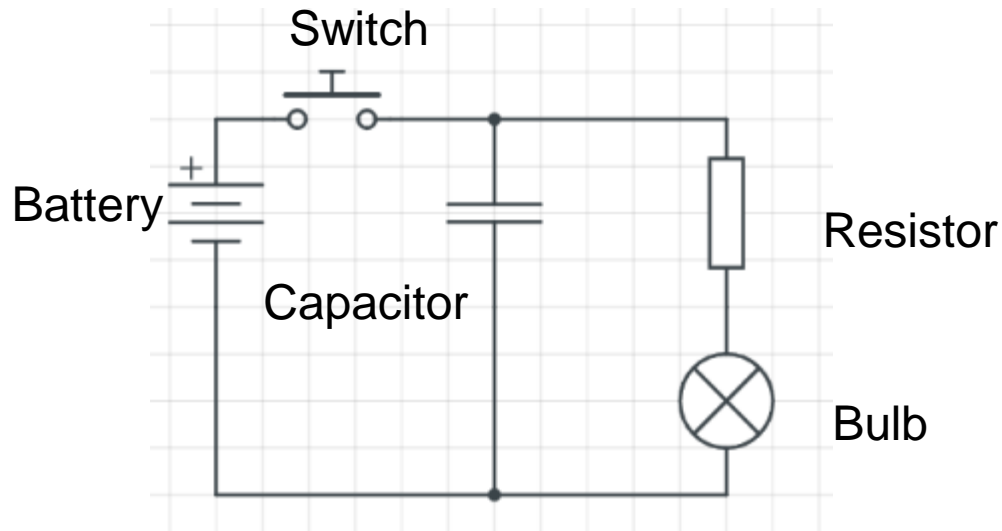


- *Given the components (Battery, wires, light bulb, resistor) build the circuit so the light bulb turns on*
- *What do you notice about the direction of flow of electrons vs the direction of flow of the current?*
- *Change the value of the resistor.
What do you notice about the current?
What do you notice about the light?*
- *Add a switch into the circuit. What do you notice if you open or close the switch?*

Circuit with Capacitor



Phet Simulator



- *Add a capacitor into the circuit and build the circuit.*
- *What do you notice about the position of the capacitor?*
- *What happens if you close the switch?*
- *What happens if you open the switch?*
- *Place the capacitor in series what do you notice?*
- *Change the display on the simulation to show the circuit diagram or schematic instead.*



Layers of a Computing System



Oide

End-user

Application Software

System Software

Computer Components

Electronics, Gates and Circuits

Data Representation

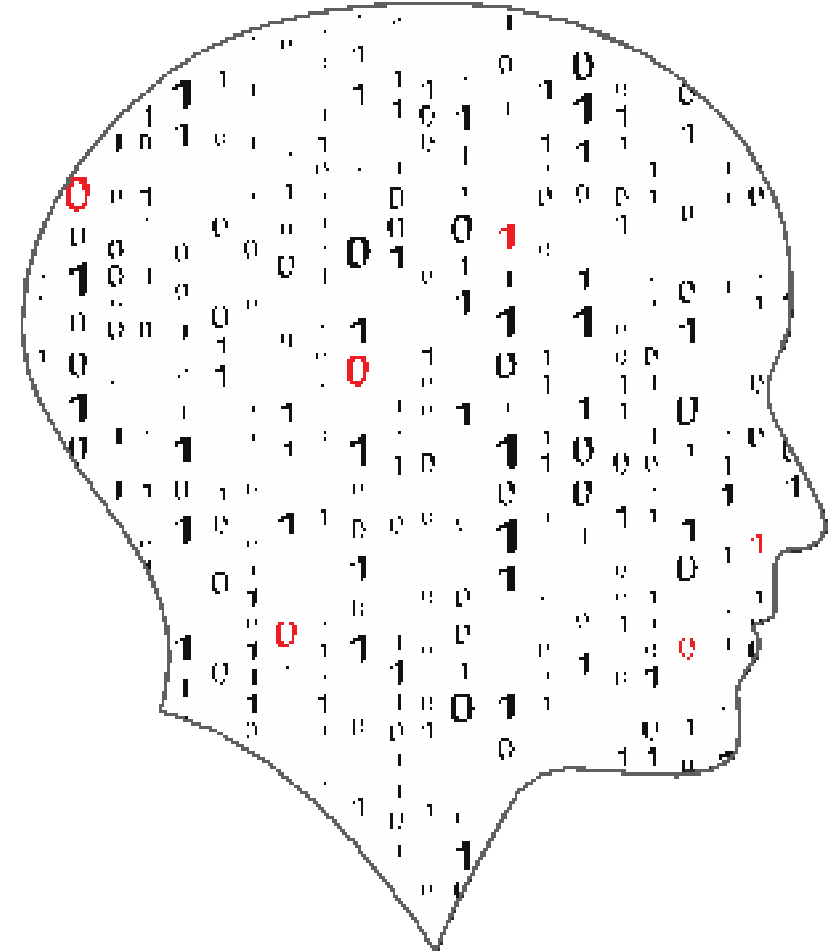


Oide

Tacú leis an bhFoghlaim
Ghairmiúil i measc Ceannairí
Scoile agus Múinteoirí

Supporting the Professional
Learning of School Leaders
and Teachers

Logic Gates



LEAVING CERTIFICATE
COMPUTER SCIENCE





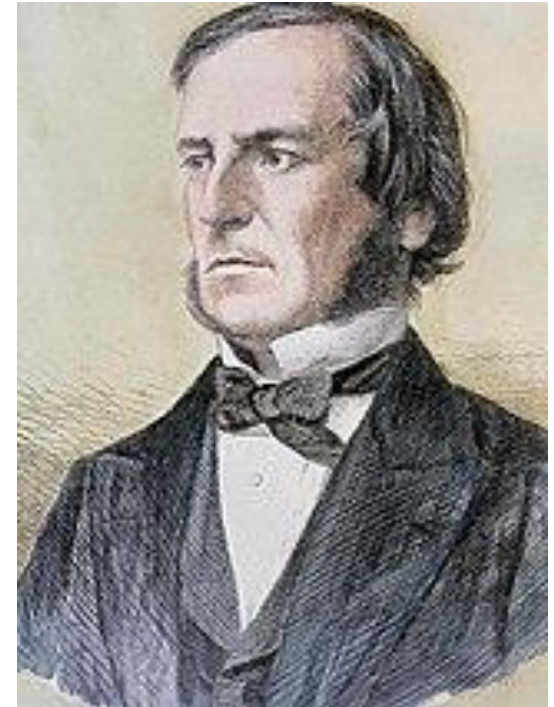
Logic Gates

A gate is a device that performs a logical operation on electrical signals.

These electrical signals are represented by bits (Binary digits) 0 (0V) or 1 (5 V/3.3 V)

The logical operations were defined by the mathematician George Boole (1815-64)

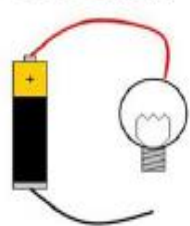
The most common logic (Boolean) operations are:



Closed circuit



Open circuit

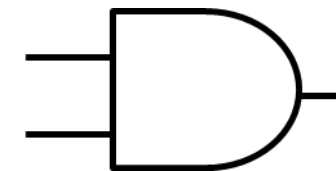
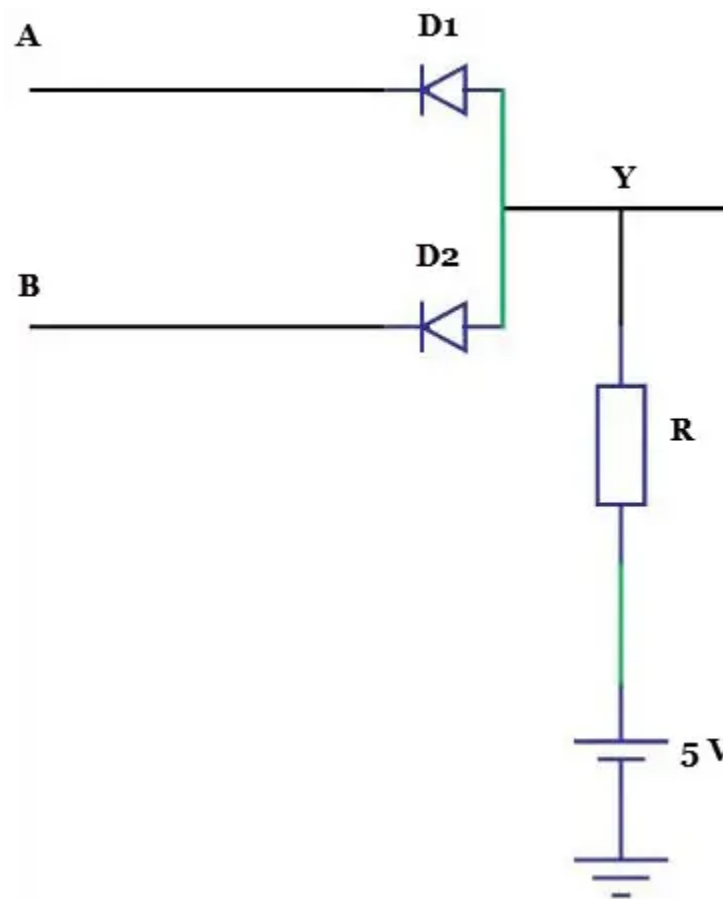
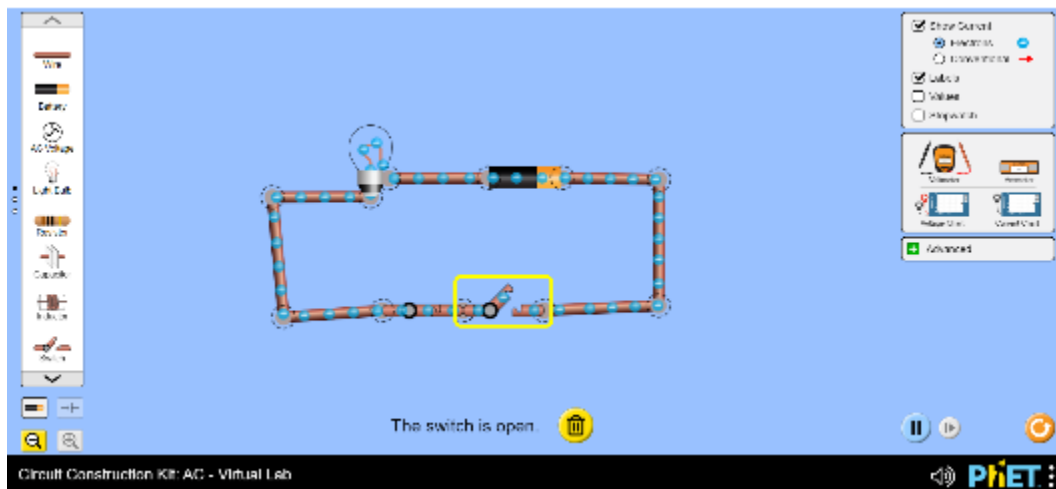


NOT
AND
OR

XOR
NAND
NOR

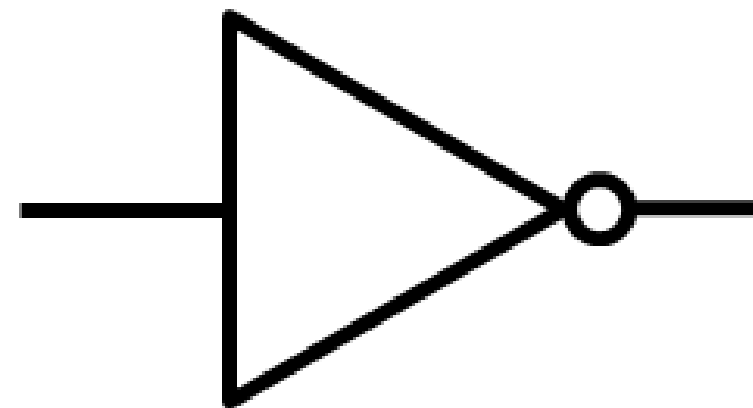
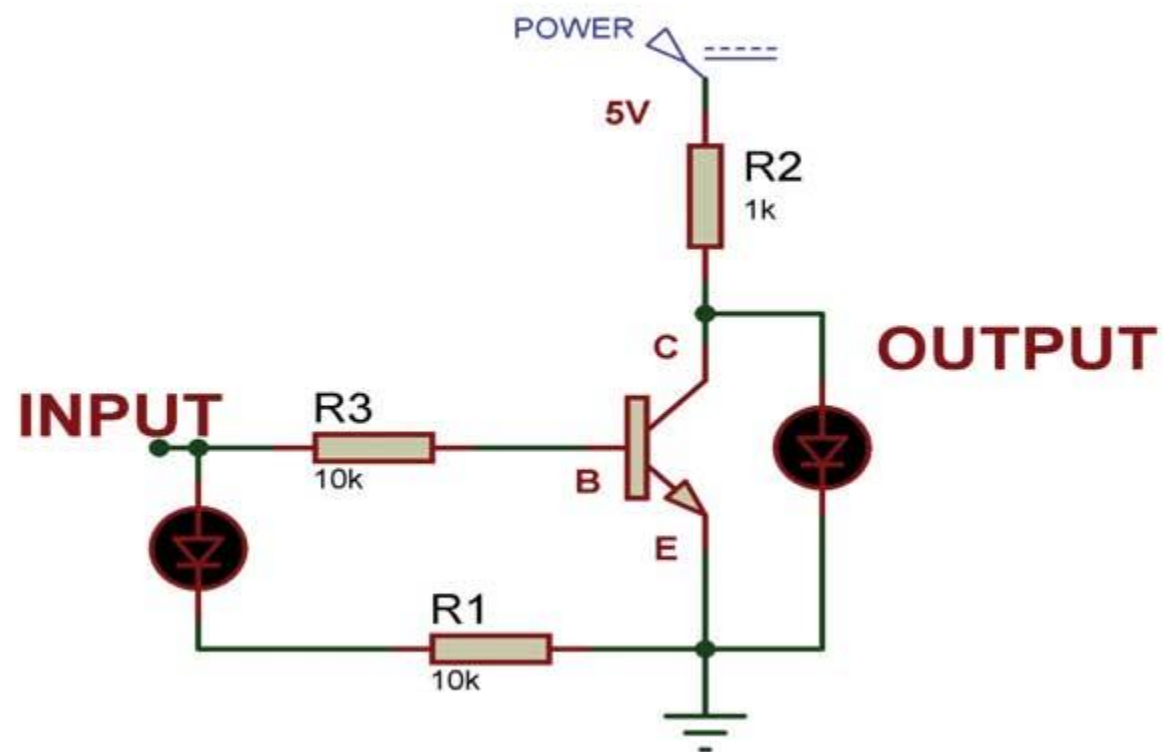


AND Gate





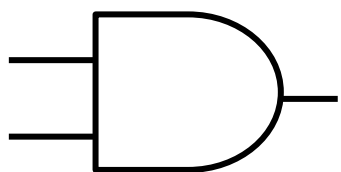
NOT Gate



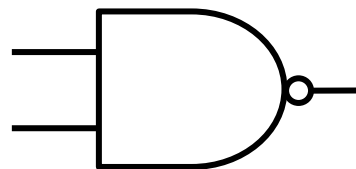


Logic Gate Symbols

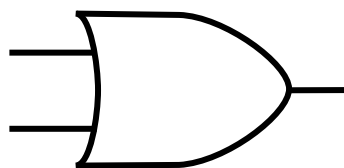
Logic gates have one or more inputs and a single output.
Each gate has its own logic symbol which allows circuits to be represented by a logic diagram



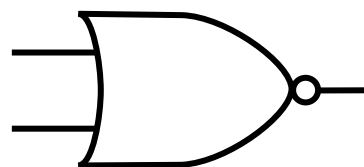
AND



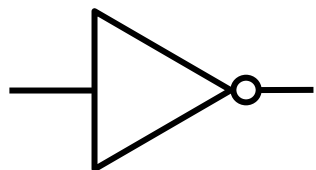
NAND



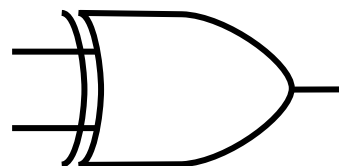
OR



NOR



NOT

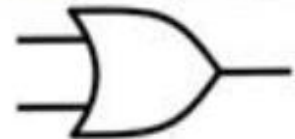
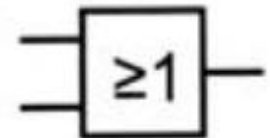
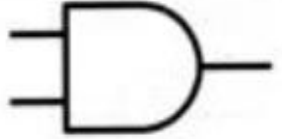
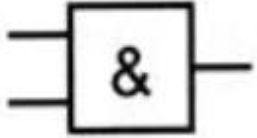
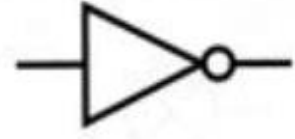
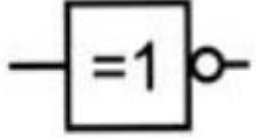
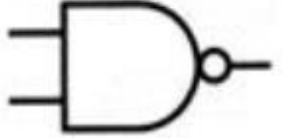
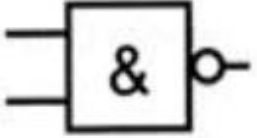
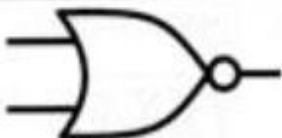
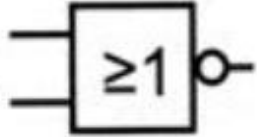
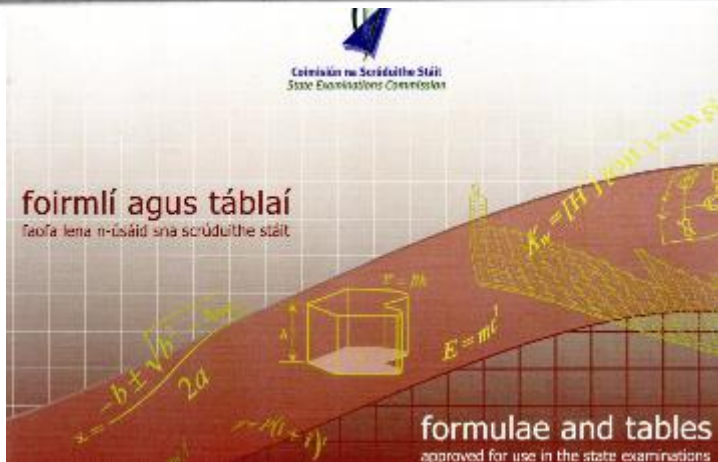


Exclusive OR (XOR)



Geataí loighce

Logic gates

  OR	  AND	  NOT (<i>inbhéartóir / inverter</i>)	  NAND
  NOR	<p>Page 78 Formula and Tables book</p> 		



Logic Diagrams, Truth tables and Boolean Expressions

The behaviour of gates (and circuits) are commonly represented in any of the following ways:

Boolean Expressions

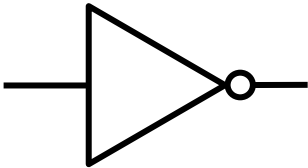
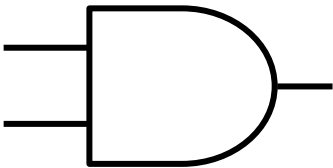
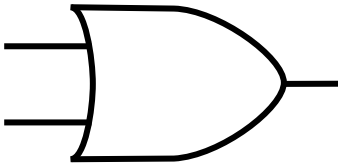
Uses Boolean algebra, a mathematical notation for expressing two-valued logic

Truth Tables

A table showing all possible input values and the associated output value

Logic Diagrams

A graphical representation of a circuit; each gate has its own symbol

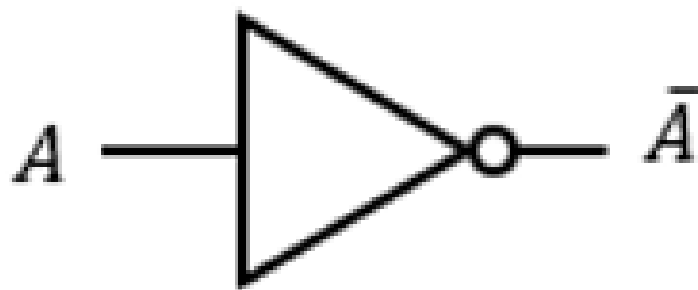
Symbol	Operator	Boolean
	NOT	\bar{A}
	AND	$A.B$
	OR	$A + B$



The NOT operation



P 18



Logic Gate Symbol

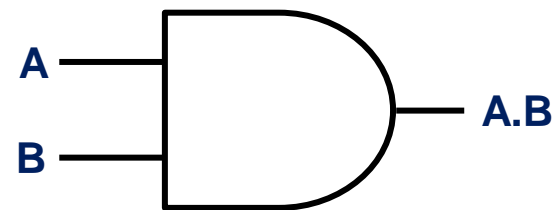
A	\bar{A}
0	1
1	0

Truth Table

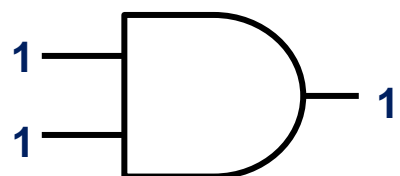
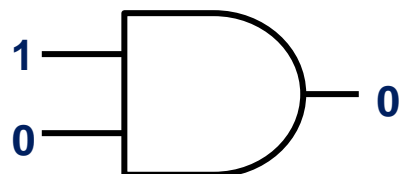
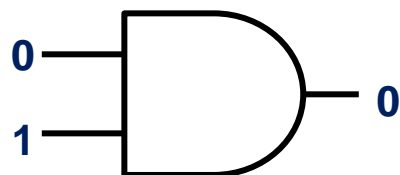
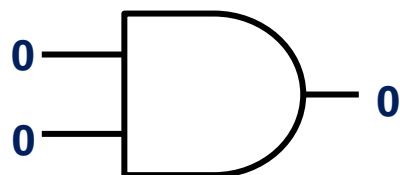
Inverts a single input. Also called an *inverter*.



The AND operation



Logic Gate Symbol



A	B	A.B
0	0	0
0	1	0
1	0	0
1	1	1

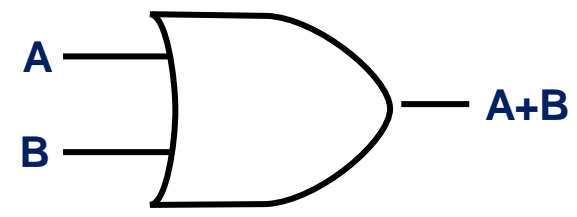
Truth Table

In order for the output to be 1 both inputs must be 1

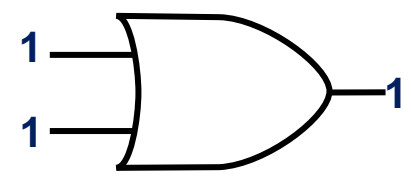
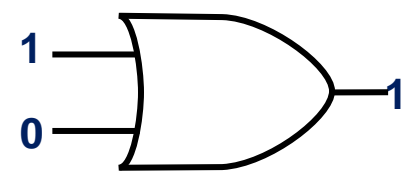
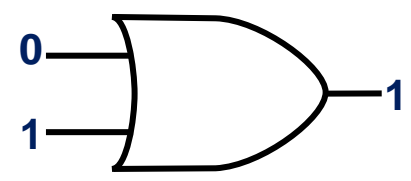
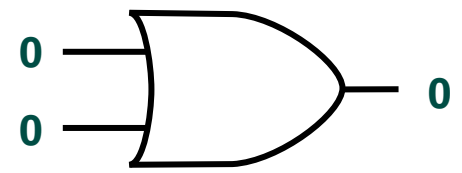




The OR operation



Logic Gate Symbol



A	B	A+B
0	0	0
0	1	1
1	0	1
1	1	1

Truth Table

In order for the output to be 1 either input must be 1

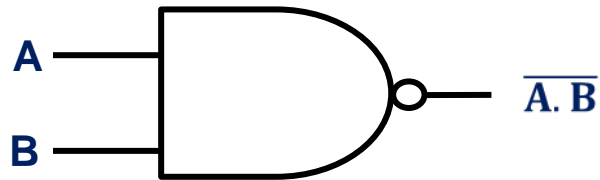




The NAND operation



P 18



Logic Gate Symbol

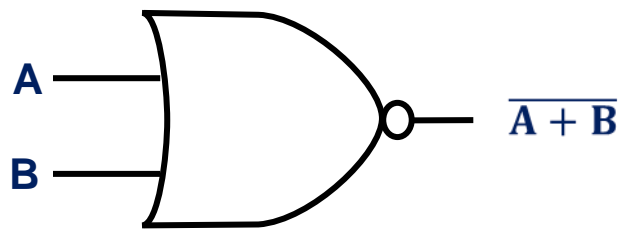
A	B	
0	0	1
0	1	1
1	0	1
1	1	0

Truth Table

The output is 1 if either input is 0



The NOR operation



Logic Gate Symbol

A	B	
0	0	1
0	1	0
1	0	0
1	1	0

Truth Table



P 18

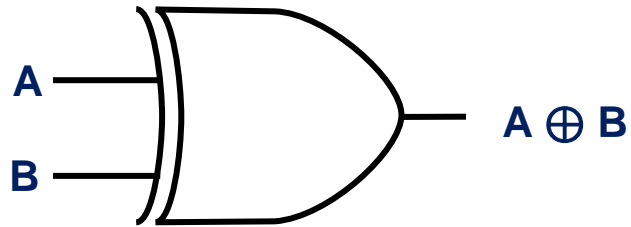
The output is 1 if both inputs are 0



The XOR operation



P 18



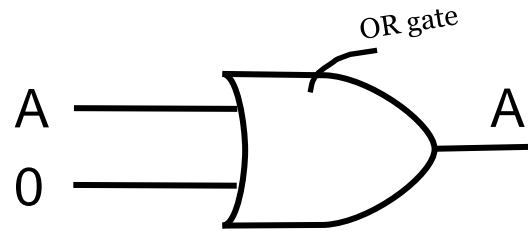
Logic Gate Symbol

A	B	
0	0	0
0	1	1
1	0	1
1	1	0

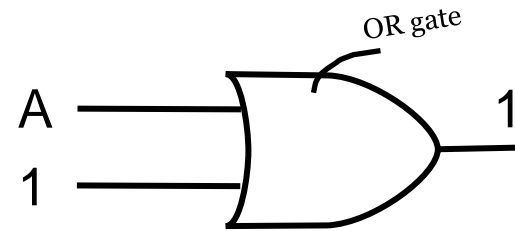
Truth Table

The output is 1 if both inputs are different

Boolean Algebra

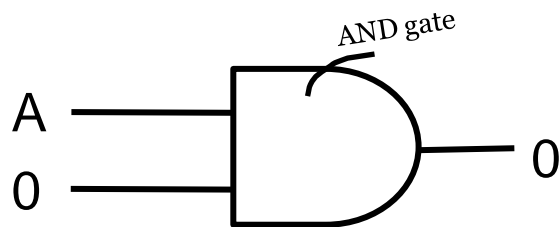


A variable *OR*ed with 0 always gives the variable

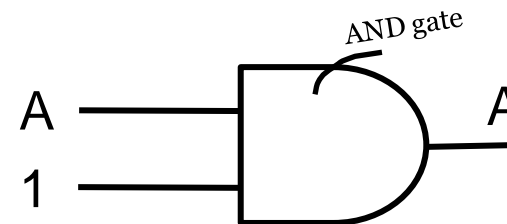


When a variable is *OR*ed with 1 the output is always 1

1. $A + 0 = A$	7. $A \cdot A = A$
2. $A + 1 = 1$	8. $A \cdot \bar{A} = 0$
3. $A \cdot 0 = 0$	9. $\bar{\bar{A}} = A$
4. $A \cdot 1 = A$	10. $A + AB = A$
5. $A + A = A$	11. $A + \bar{A}B = A + B$
6. $A + \bar{A} = 1$	12. $(A + B)(A + C) = A + BC$



When a variable is *AND*ed with 0 the output is always 0.

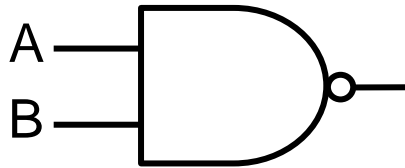


Here we can see that when a variable is *AND*ed with 1 the output is always the variable

Using truth tables to verify identities



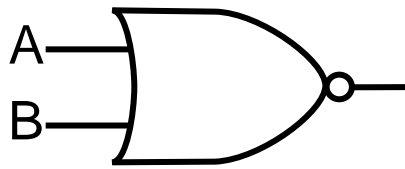
$$\overline{A \cdot B} = \bar{A} + \bar{B}$$



A NAND B

A	B	A · B	$\overline{A \cdot B}$	\bar{A}	\bar{B}	$\bar{A} + \bar{B}$
0	0	0	1	1	1	1
0	1	0	1	1	0	1
1	0	0	1	0	1	1
1	1	1	0	0	0	0

$$\overline{A + B} = \bar{A} \cdot \bar{B}$$



A NOR B

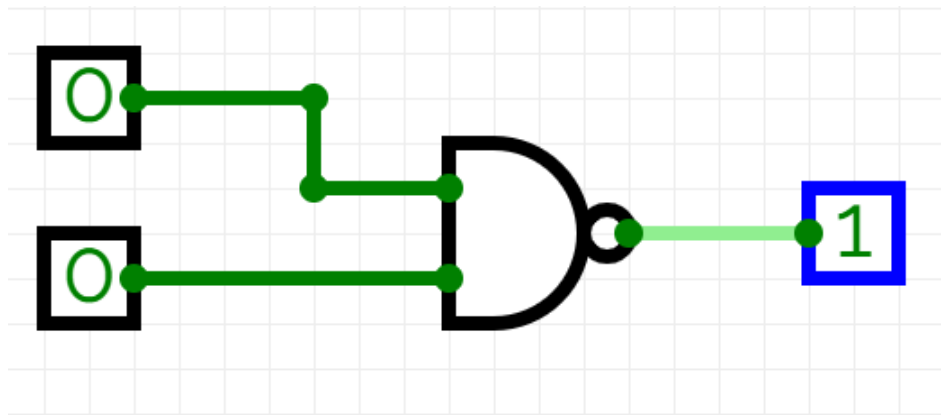
A	B	A + B	$\overline{A + B}$	\bar{A}	\bar{B}	$\bar{A} \cdot \bar{B}$
0	0	0	1	1	1	1
0	1	1	0	1	0	0
1	0	1	0	0	1	0
1	1	1	0	0	0	0



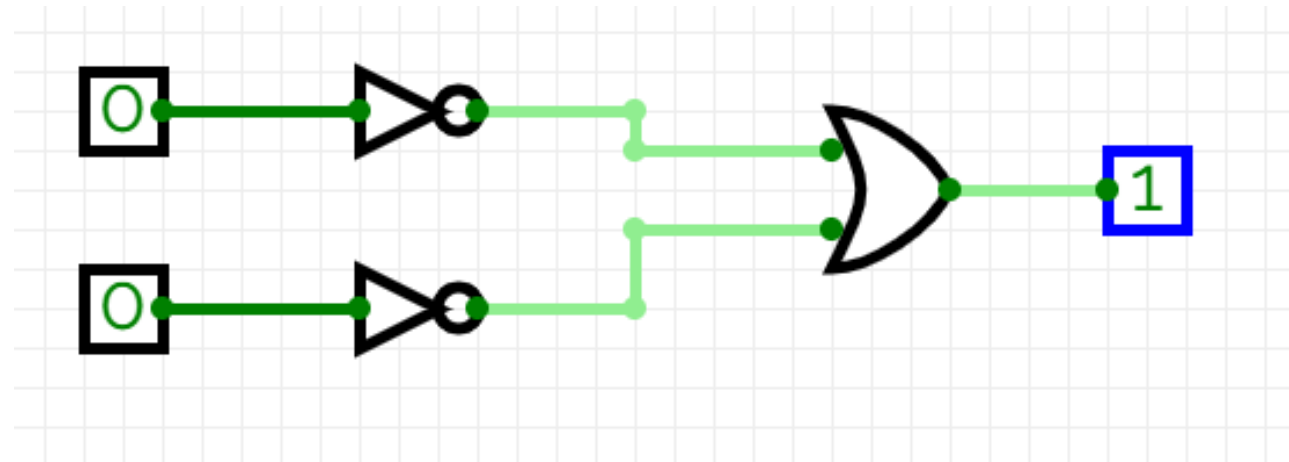
Using Logic Gates to verify identities

$$\overline{A \cdot B} = \bar{A} + \bar{B}$$

$$\overline{A \cdot B}$$



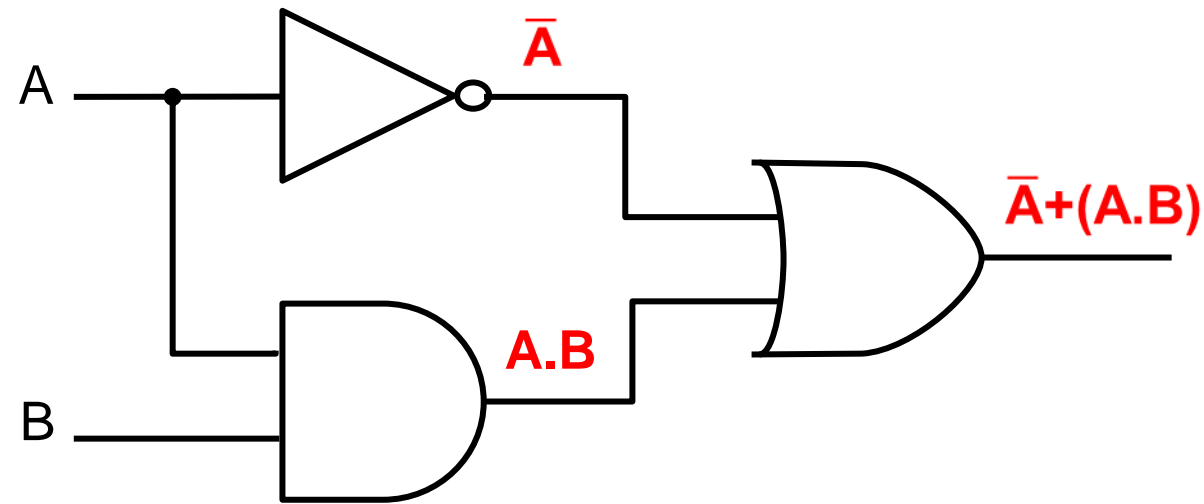
$$\bar{A} + \bar{B}$$





Connecting Logic Gates

Logic gates may be combined by using the output of one gate as the input to another.



Work progressively from the inputs to the output ...adding logic expressions to the output of each gate in turn

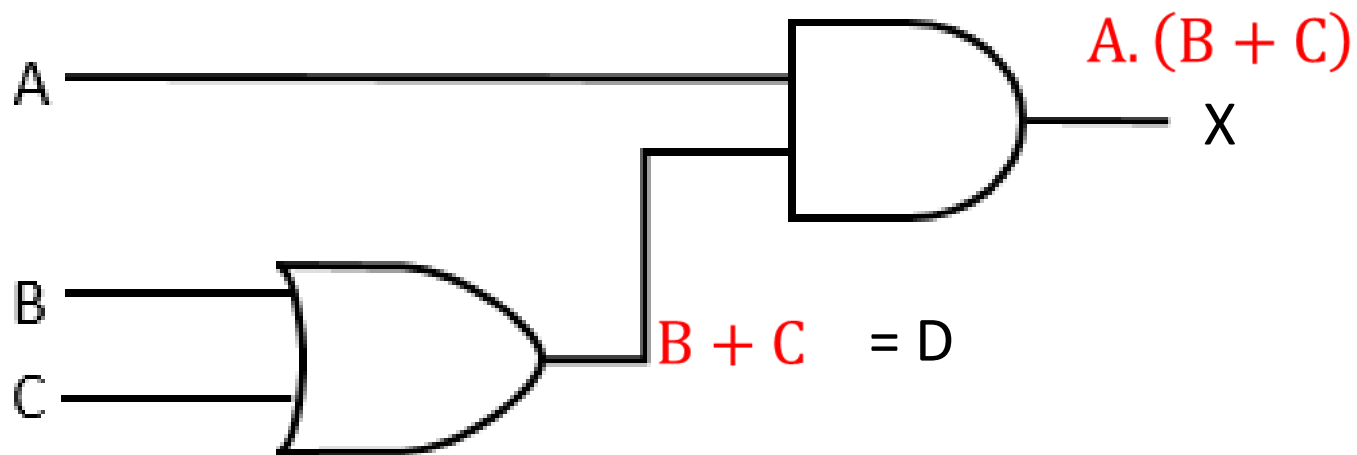


Connecting Logic Gates to create circuits

Consider this circuit used to model a burglar alarm. Once the alarm is ON (input A) then either a PIR sensor (input B) or a sound sensor (PIR B) can trigger the alarm.



P 20



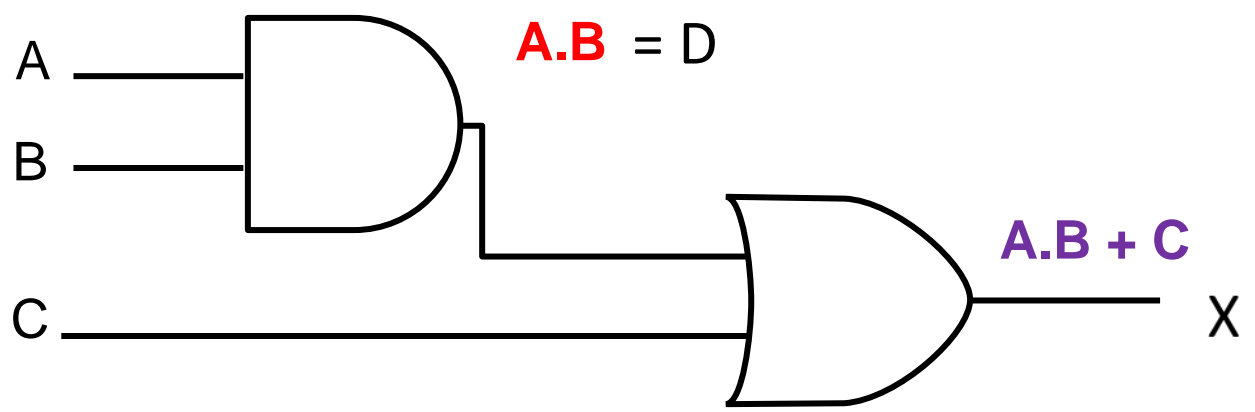
A	B	C	D	X
0	0	0	0	0
0	0	1	1	0
0	1	0	1	0
0	1	1	1	0
1	0	0	0	0
1	0	1	1	1
1	1	0	1	1
1	1	1	1	1

What about a truth table for the above circuit?



Connect Logic Gates (to create circuits)

Consider this circuit used to model a smart light. The light comes on if it is dark (input A) and it detects motion (input B) or it is switched on manually (C).



A	B	C	D	X
0	0	0	0	0
0	0	1	0	1
0	1	0	0	0
0	1	1	0	1
1	0	0	0	0
1	0	1	0	1
1	1	0	1	1
1	1	1	1	1





Write code for basic gates

Write a function in a programming language for one of the basic logic gates:

- AND
- OR
- XOR
- NAND
- NOR

The function should take 2 inputs (A,B) and output the result from the gate.



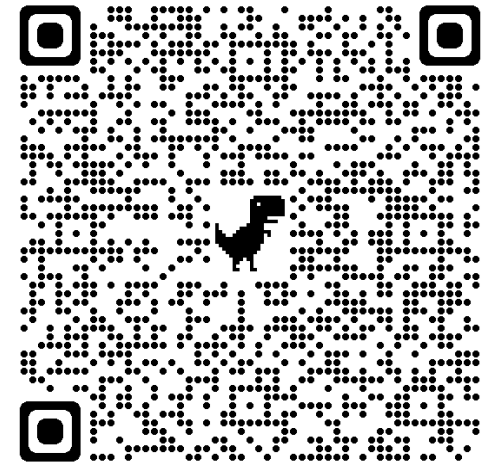
Group Task

Task: Design a Half-Adder
(NCCA booklet p23)

Design your own half-adder in some language you have learned, for example Python or Scratch

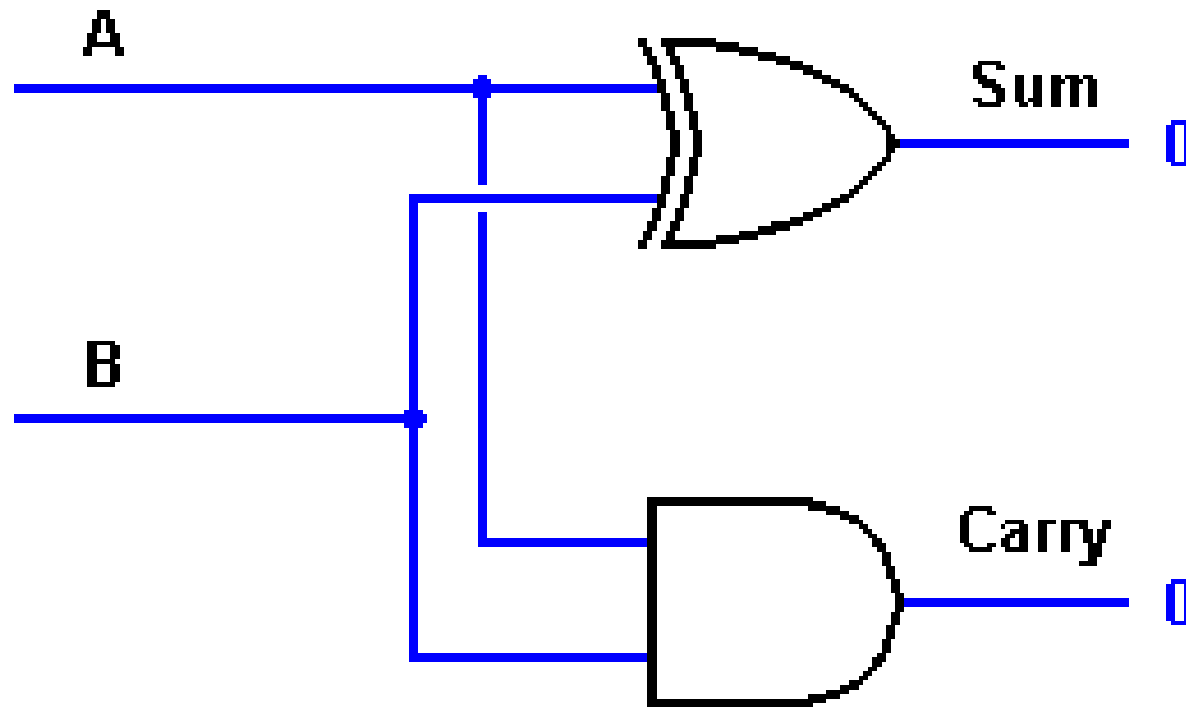
INPUTS		OUTPUT	
A	B	Sum (S)	Carry (C)
0	0	0	0
0	1	1	0
1	0	1	0
1	1	0	1

With the help of half adder, we can design circuits that are capable of performing simple addition with the help of logic gates.





Half Adder





Group Activity

In your groups, allocate the following tasks (one task per person preferably):

- build the logic gate circuitry with pen & paper
- build the logic gate circuit using [CircuitVerse software](#)
- write the Boolean expressions
- write the Python code for the logic gate circuit



Present your findings to the other groups.



An Roinn Oideachais
Department of Education





Oide

Tacú leis an bhFoghlaim
Ghairmiúil i measc Ceannairí
Scoile agus Múinteoirí

Supporting the Professional
Learning of School Leaders
and Teachers

National Workshop 4

Session 4

Introduction to Modelling and Simulation





Key Messages for National Workshop 4



There are many ways to use the LCCS specification.



The study of Computers and Society is one of the overarching principles of LCCS.



LCCS can be mediated through a constructivist pedagogical approach.

ALTs

ALTs provide an opportunity to teach theoretical aspects of LCCS.



Critical reflection will be a central component of the student experience and the LCCS teacher's PD journey.



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



By the end of this session participants will have the opportunity to:



enhance their understanding of modelling, simulation and ALT 3



collaborate with others to **design one** potential solution for ALT 3, as well as give and receive feedback on potential ALT 3 designs



explore a variety of models from online resources and Python code provided



reflect on their learning and ways by which learning outcomes from ALT 3 might be linked with other aspects of the course

Session Overview



Section 1	Introduction to Modelling and Simulation
Section 2	Group Activities and Discussion
Section 3	Agent Based Modelling / Complex Systems



Key



Booklet



Individual Activity



Group Activity



Reflection



Group Feedback

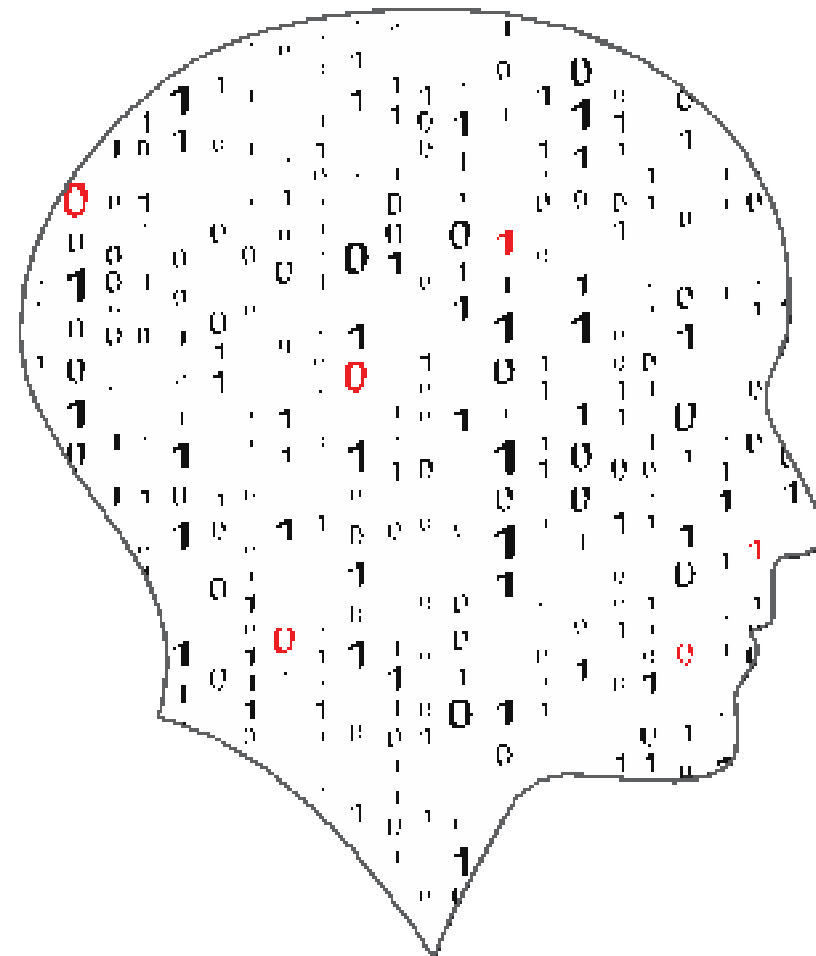


Oide

Tacú leis an bhFoghlaim
Ghairmiúil i measc Ceannairí
Scoile agus Múinteoirí

Supporting the Professional
Learning of School Leaders
and Teachers

Section I Introduction to Modelling and Simulation



LEAVING CERTIFICATE
COMPUTER SCIENCE





Activity 1 (Think)



Think

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



Pair

Participants turn to the person beside them to discuss their ideas



Share

Pairs share their answers with other pairs (square) or the wider group

THINK!

Watch the following video and jot down your thoughts in your workbook.





Activity 1 (Pair and Share)



Think

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



Pair

Participants turn to the person beside them to discuss their ideas



Share

Pairs share their answers with other pairs (square) or the wider group

PAIR AND SHARE!

What is a computer model?

Agree 3 - 5 examples of computer models.





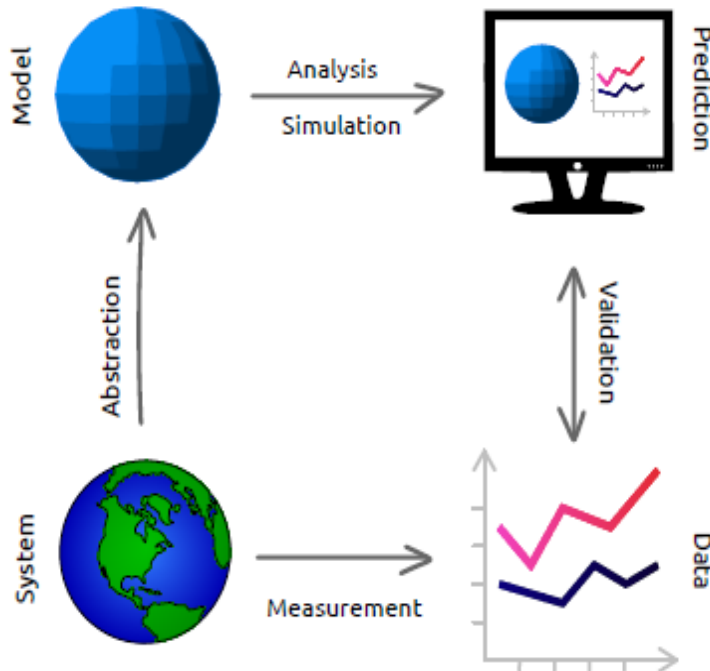
Modelling and Simulation at Pixar



Source: <https://www.khanacademy.org/partner-content/pixar/simulation/hair-simulation-101/v/hair-simulation-intro>



Modelling and Simulation



Models are systems created to imitate parts of the real world; systems which can be understood, manipulated and examined more easily.

A **simulation** is a process of running a computer program that uses the model to simulate the behaviour of the system.

Modelling vs. Simulation

Simulations require the use of models; the model represents the key characteristics or behaviours of the selected system or process, whereas the simulation represents the evolution of the model over time. (<https://en.wikipedia.org/wiki/Simulation>).

Source: <https://greenteapress.com/wp/modsimpy/>

- ★ A model is a representation of a real-world system or situation.
- ★ A simulation shows what happens when a model is applied over time.



Types of Models

Physical

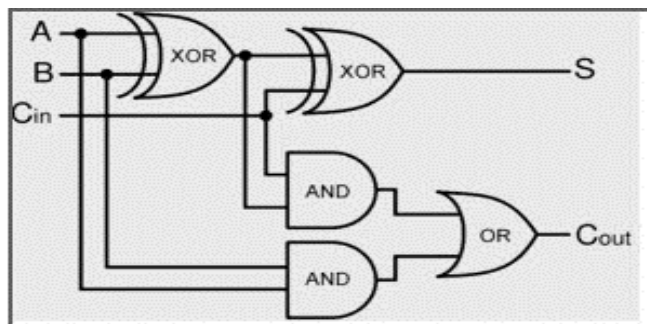


Mathematical



P. 22

Schematic

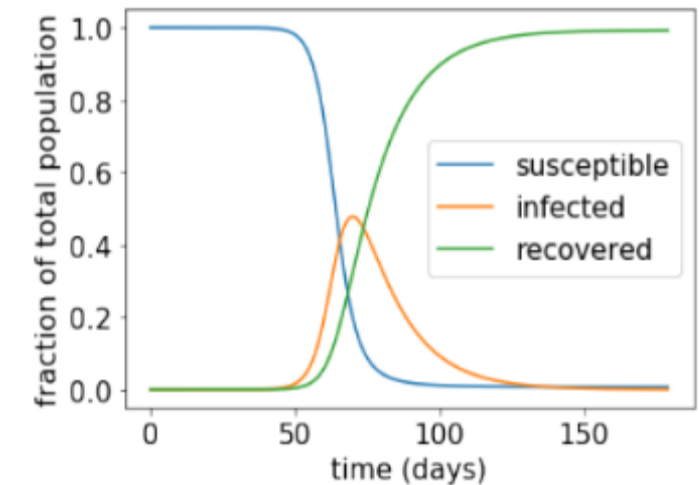
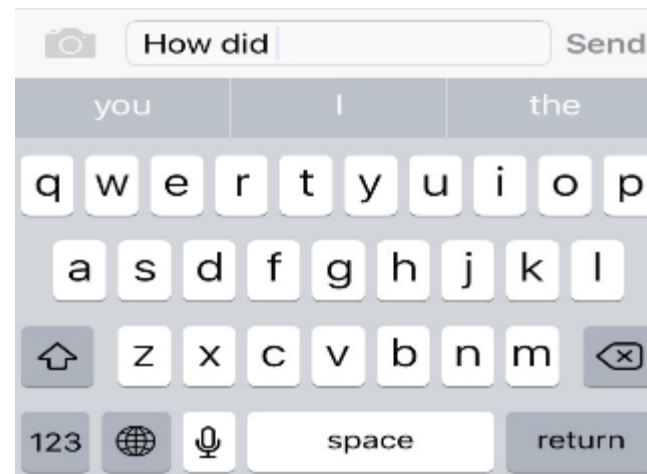
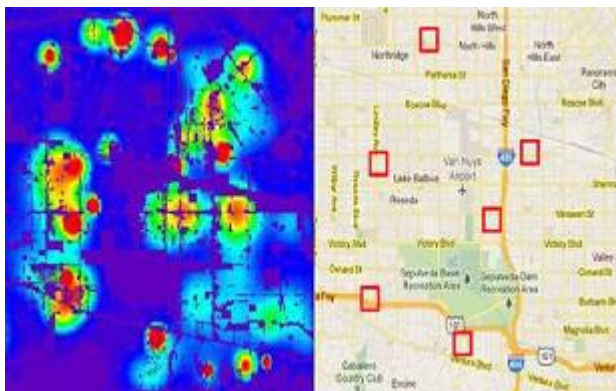
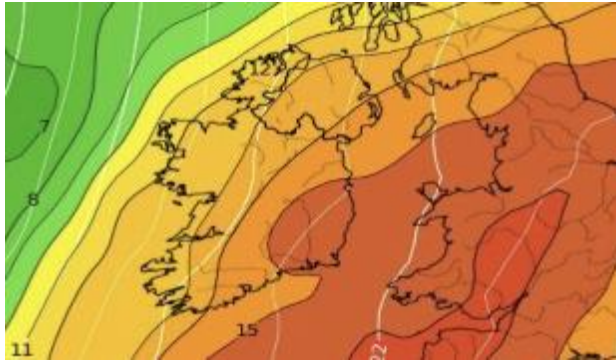


Computational





Examples of Computational Models





Benefits of Modelling

- ✓ **Test feasibility**
- ✓ **Gain insights/explore 'what if' questions**
- ✓ **Make predictions**
- ✓ **Cost/Safety**
- ✓ **Gaining competitive advantage**



A Simple Fish Population Model

Problem statement:
Is the fish population sustainable?

Variables:
initial fish population,
growth rate, allowable
harvest number of
years

```
import matplotlib.pyplot as pyplot

# A simple population model
initial_population = int(input("Enter the starting population: ")) # e.g. 4000000
years = int(input("Enter the number of years: ")) # e.g. 10 or 100
growth_rate = int(input("Enter the percentage growth rate: ")) # e.g. 8
growth_rate = growth_rate/100
harvest = int(input("Enter the maximum annual harvest allowed: ")) # e.g. 1500

print("Year \t Population")
print("==== \t =====")
population = initial_population
population_list = [ ]
population_list.append(initial_population)
for year in range(years):
    population = (1+growth_rate) * population - harvest
    population_list.append(population)
    print(year + 1, "\t", int(population))

print('The final population is %.2f' %population)

# Display the results graphically
pyplot.plot(range(years + 1), population_list)
pyplot.xlabel('Year')
pyplot.ylabel('Population')
pyplot.show()
```



A Simple Fish Population Model



Problem statement: Is the fish population sustainable?

Variables: initial fish population, growth rate, allowable harvest
number of years

Enter the starting population: 12000

Enter the number of years: 10

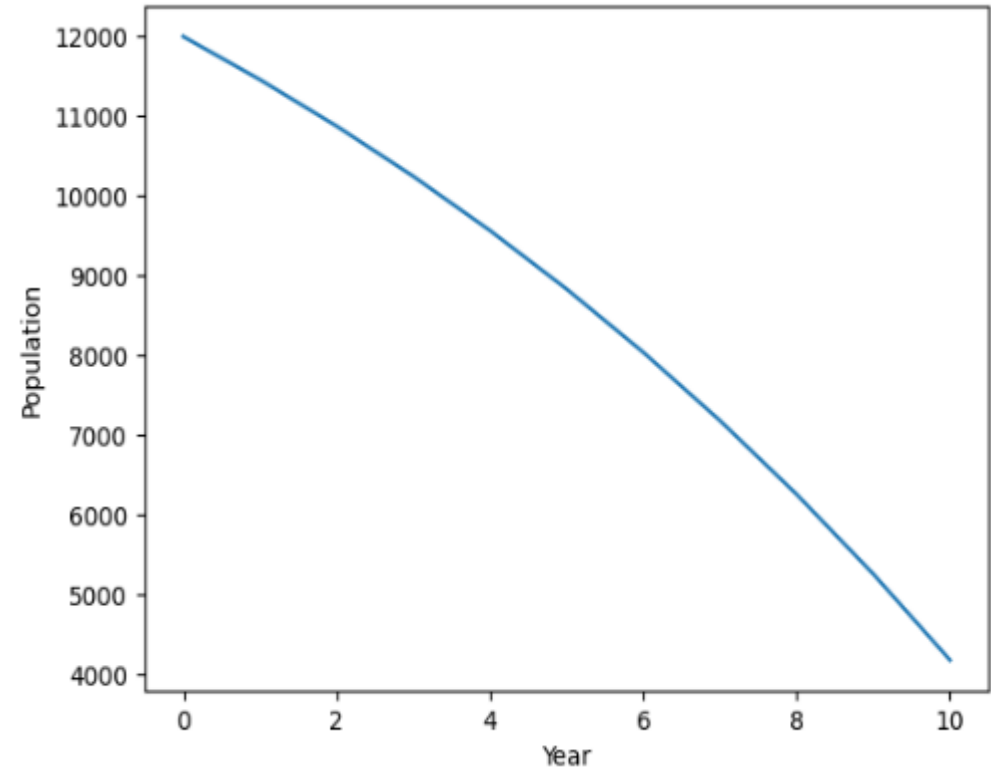
Enter the percentage growth rate: 8

Enter the maximum annual harvest

allowed: 1500

Year	Population
1	11460
2	10876
3	10246
4	9566
5	8832
6	8038
7	7181
8	6256
9	5256
10	4177

The final population is 4177.26





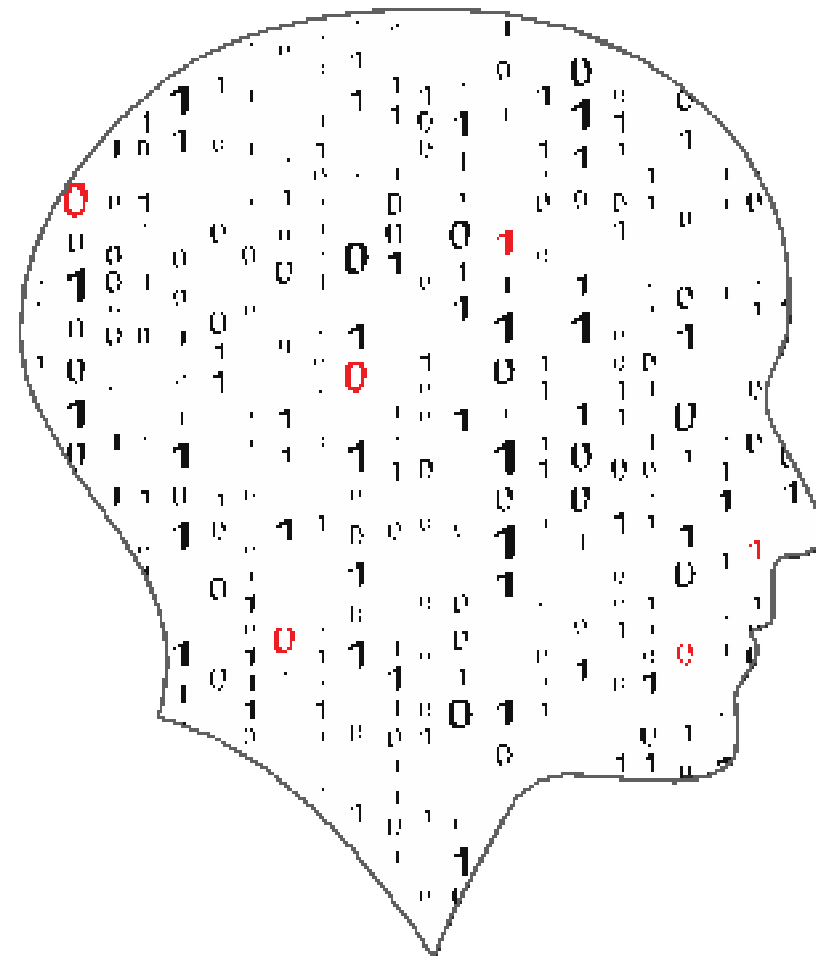
Oide

Tacú leis an bhFoghlaim
Ghairmiúil i measc Ceannairí
Scoile agus Múinteoirí

Supporting the Professional
Learning of School Leaders
and Teachers

Section II

Group Activity on Predictive Models



LEAVING CERTIFICATE
COMPUTER SCIENCE





Predictive Models

Devise a model to predict driving test outcome

ID	GENDER	AGE	CAR MODEL	OUTCOME
1	Male	17	Audi	FAIL
2	Female	28	Toyota	PASS
3	Female	20	VW	PASS
4	Male	18	Toyota	FAIL
5	Male	19	Renault	PASS
6	Male	18	Renault	FAIL
7	Female	17	Toyota	FAIL
8	Male	25	BMW	PASS
9	Female	19	VW	PASS
10	Female	23	Nissan	PASS

**IF AGE \geq 19 THEN
 OUTCOME is
 PASS
ELSE
 OUTCOME is FAIL**



Predictive Models

ID	GENDER	AGE	CAR MODEL	OCCUPATION	INCOME	OUTCOME
1	Male	17	Audi	Student	NIL	FAIL
2	Female	28	Toyota	Teacher	43210	PASS
3	Female	20	VW	Student	5500	PASS
4	Male	18	Toyota	Student	NIL	FAIL
5	Male	19	Renault	Trainee Nurse	18250	PASS
6	Male	18	Renault	Student	NIL	FAIL
7	Female	17	Toyota	Student	8300	FAIL
8	Male	25	BMW	Vet	32750	PASS
9	Female	19	VW	Apprentice	21460	PASS
10	Female	23	Nissan	Nurse	28700	PASS
11	Male	22	BMW	Apprentice	21000	PASS
12	Male	20	Nissan	Student	NIL	FAIL
13	Female	17	Honda	Student	9200	FAIL
14	Male	19	Honda	Student	NIL	FAIL
15	Male	24	Audi	Teacher	38700	PASS
16	Female	19	Renault	Student	8800	PASS
17	Female	22	BMW	Musician	50000	PASS

IF AGE < 18 THEN

OUTCOME is FAIL

ELIF AGE >= 22 THEN

OUTCOME is PASS

ELIF GENDER = 'Female'

AND AGE >=18 THEN

OUTCOME is PASS

ELSE

OUTCOME is FAIL



Activity 1: To create a predictive model based on your assigned dataset



INSTRUCTIONS

1. Decide on a target variable (this is what you are trying to model/predict).
2. Decide on the which features from the dataset to use to create your model.
3. Analyse the dataset for patterns.
4. Create your model.
5. Evaluate your model (it does not have to work 100% of the time).

	Dataset
Groups 1 & 2	World happiness
Groups 3 & 4	IMDb Top 100 Movies
Groups 5 & 6	FIFA World Cup 2022
Groups 7 & 8	Significant Earthquakes

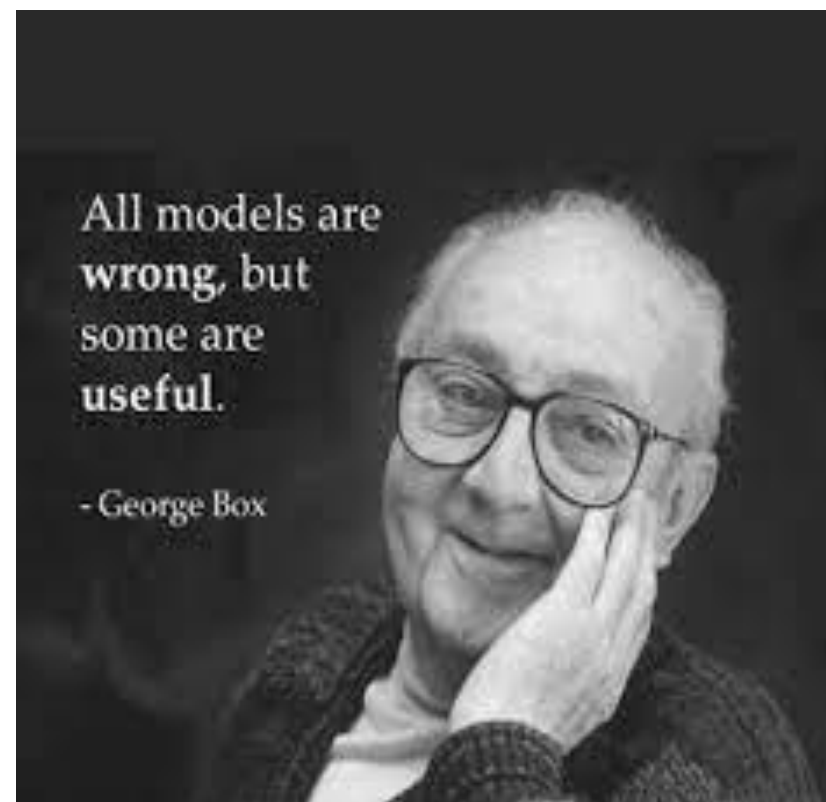


Discussion

How do computer models work?

Do they have one or more of the following?

- Messy
- Real-life
- Not easily solved otherwise
- In the future – difficult to predict / forecast
- Involve assumptions
- Involve probabilities / random





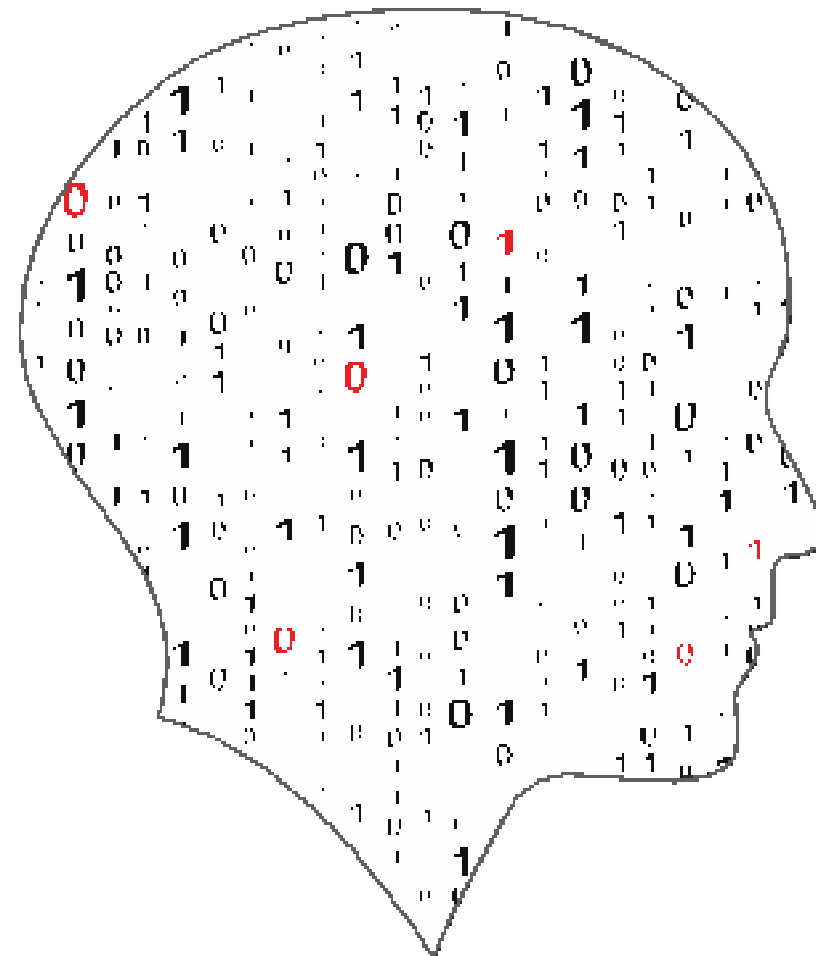
Oide

Tacú leis an bhFoghlaim
Ghairmiúil i measc Ceannairí
Scoile agus Múinteoirí

Supporting the Professional
Learning of School Leaders
and Teachers

Section II

Group activity on Random Models



LEAVING CERTIFICATE
COMPUTER SCIENCE





Activity 2: Looking at the cat and mouse model

In your groups, explore the code provided and answer the questions provided/complete the tasks.

Nominate a chair, a notetaker, a timekeeper and a spokesperson (to provide feedback).

Present your findings to the other groups.





Instructions

Download the code from GitHub.

What could this model be used for?

What are its limitations?

How could the model be improved?

How could we extend the model?

How could this model be adapted to simulate multiple different scenarios? Graph the outcomes.



P. 28



15 minutes



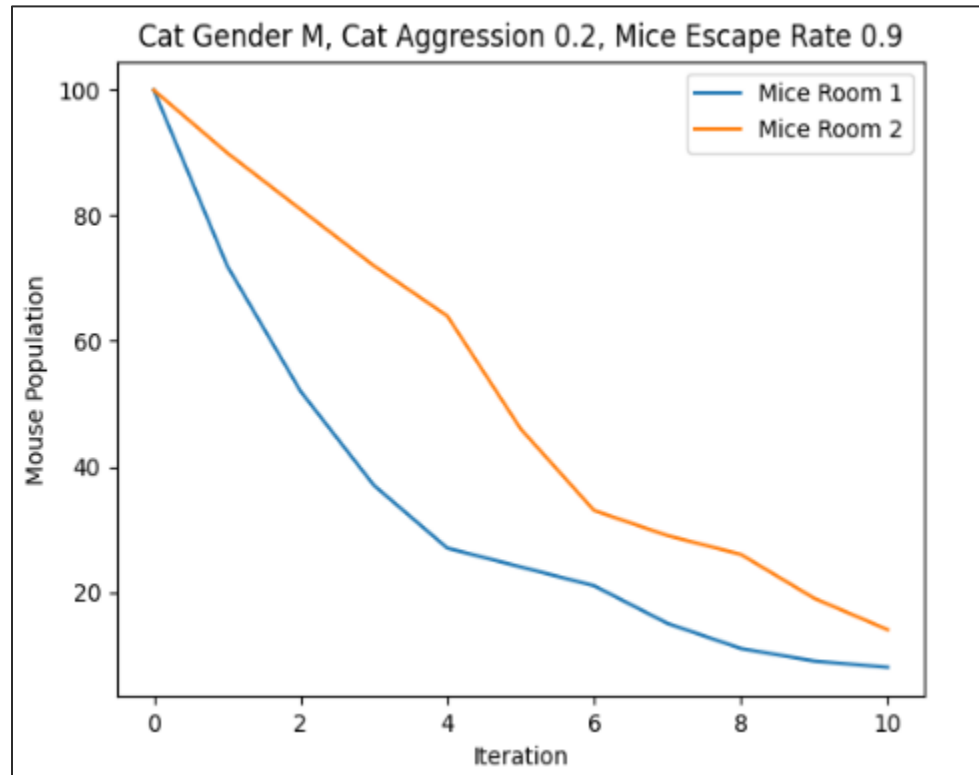
Group Activity: Feedback



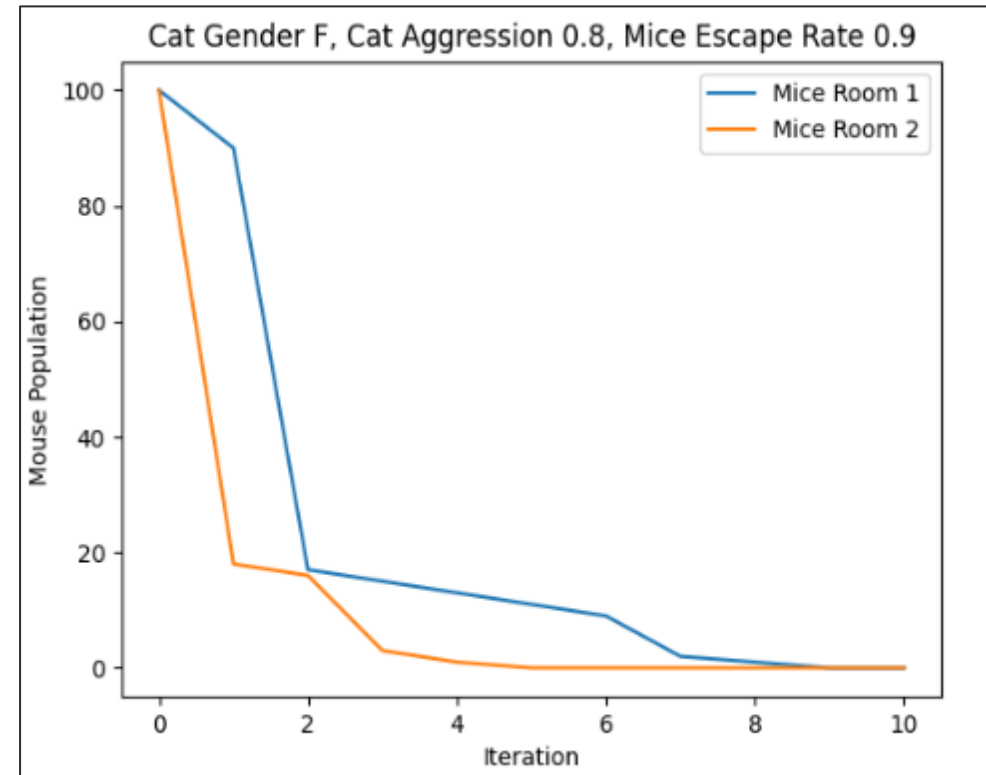


Cat and Mice Model (version 1)

Variables: initial mouse population, cat gender, cat aggression level, mouse escape rate



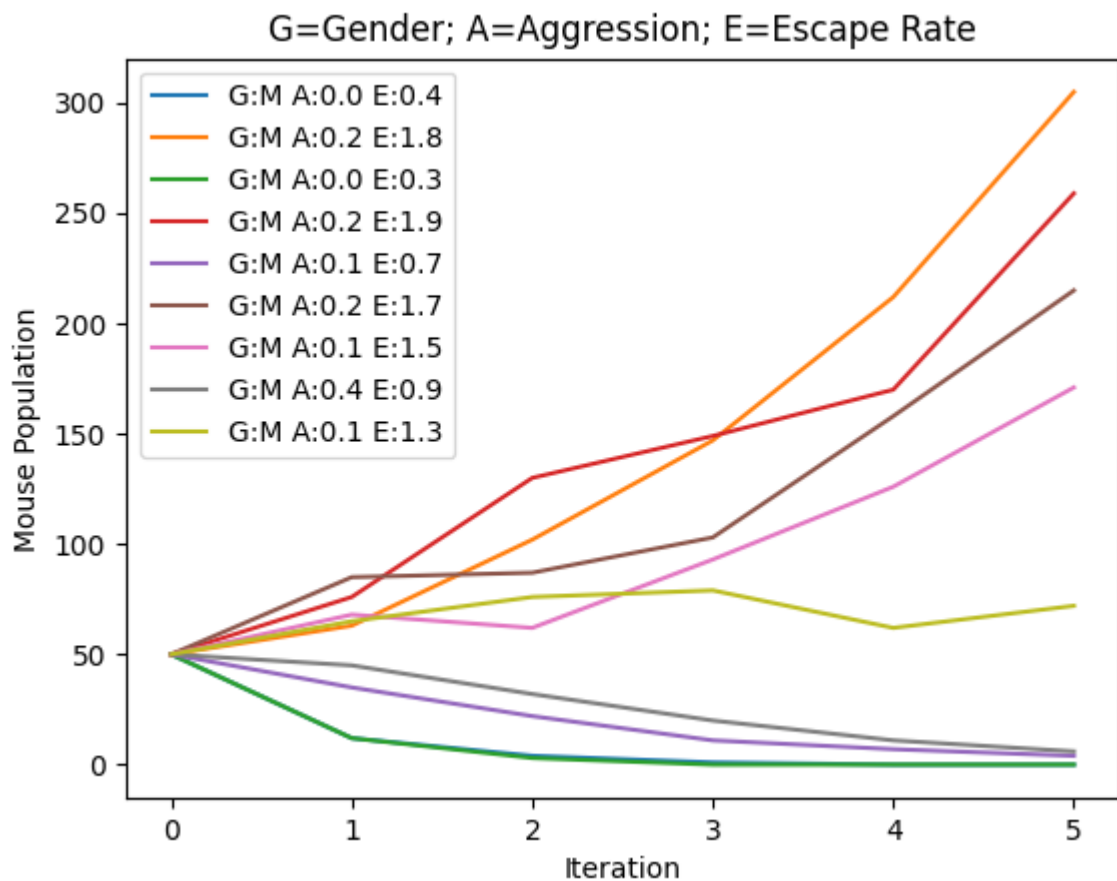
Scenario #1



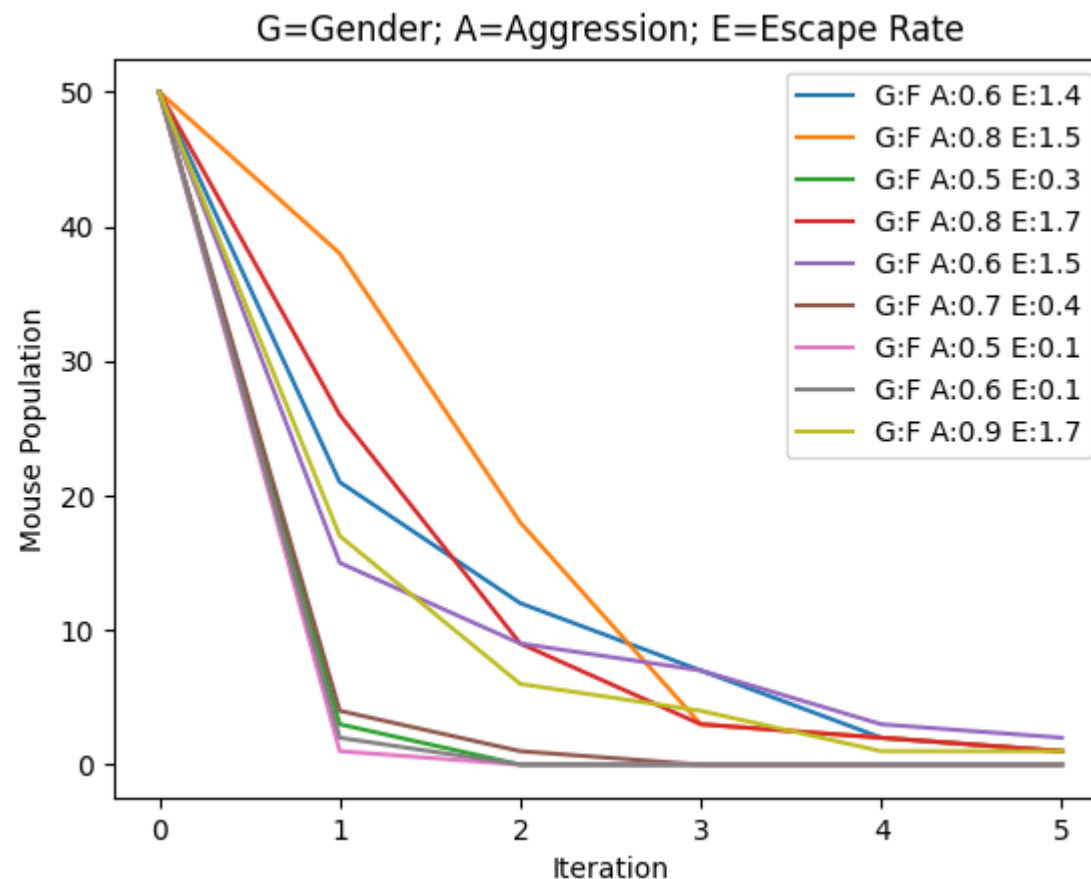
Scenario #2



Cat and Mice Model (version 2)



Scenario #1



Scenario #2

Summary



Modelling

The process by which phenomena or objects can be represented (a good example of abstraction).

A theoretical framework from which past performances can be used to make predictions about the future.

Can be physical or abstract (mathematical or computational).

Abstract Model



Complex Reality



Simulations

Show what happens when a model is run.

A third pillar of science (in addition to theory and experimentation).

<http://greenteapress.com/complexity2/html/thinkcomplexity2010.html>



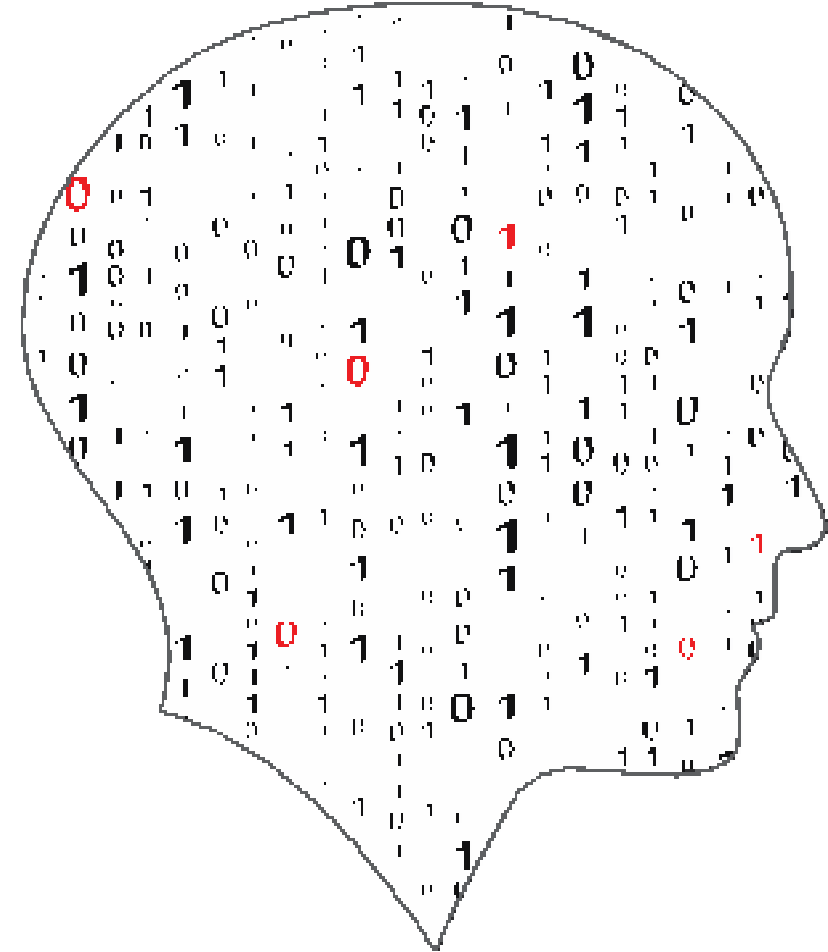
Oide

Tacú leis an bhFoghlaim
Ghairmiúil i measc Ceannairí
Scoile agus Múinteoirí

Supporting the Professional
Learning of School Leaders
and Teachers

Section III

Agent Based Modelling (ABM) / Complex Systems



LEAVING CERTIFICATE
COMPUTER SCIENCE





ALT 3: Learning Outcomes

3.8 develop a model that will allow different scenarios to be tested

3.9 analyse and interpret the outcome of simulations **both before and after modifications have been made**

3.10 explain the benefits of using agent-based modelling and how it can be used to demonstrate emergent behaviours



Agent based modelling (ABM)



<https://youtu.be/R4lov0SaFpl>

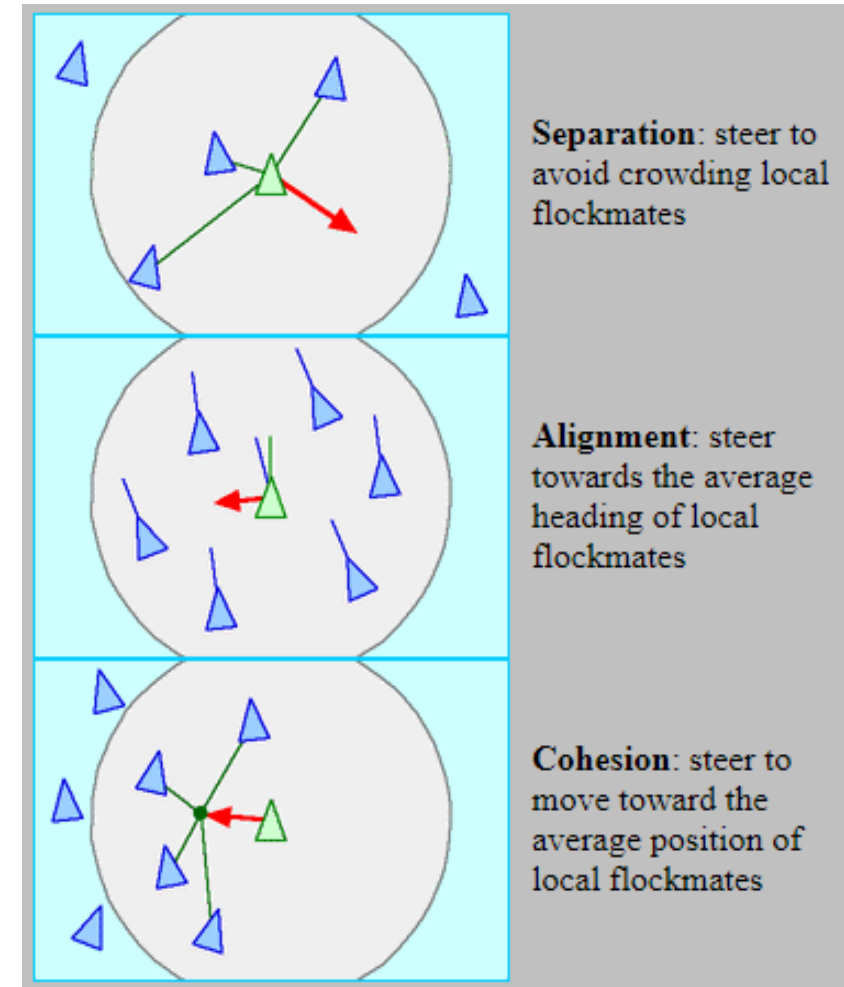


Agent Based Modelling

- A kind of modelling used to explore the behaviour of large-scale complex systems which are made up of autonomous individuals called agents
- Each agent has a set of properties which change/can be modified
- Because the agents are autonomous the values of the properties for each individual agent can differ.
- The behaviour of the overall system is determined by the collective (as opposed to the individual) behaviour

Emergent Behaviours

- A characteristic of a system that results from the interaction of its components.
- Defining characteristic may be unpredictability.





Oide

If modelling a contagion with one person sick, doing a random walk amongst the population, then the model can demonstrate how that contagion spreads to the entire population.

If, instead of a random walk, the people behaved differently- congregated at the local supermarket for example and exercised social distancing otherwise - then the contagion's behaviour will differ too, spread slower hopefully.

If small fish for instance swam around randomly until they came close to another small fish. There after the two fish stayed together, collecting other small fish. What would emerge is a school of little fish, swimming around together, pretending to be one great big fish

If a population had to compete for a scarce food supply, then those who are faster/stronger might get that food for themselves, the weaker would starve and die off, and the food supply might become sufficient for the reduced population.



Online Resources



Netlogo



Project GUTS



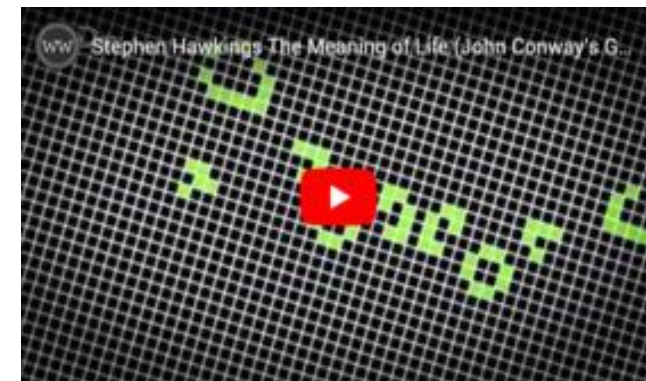
PhET



Pixar



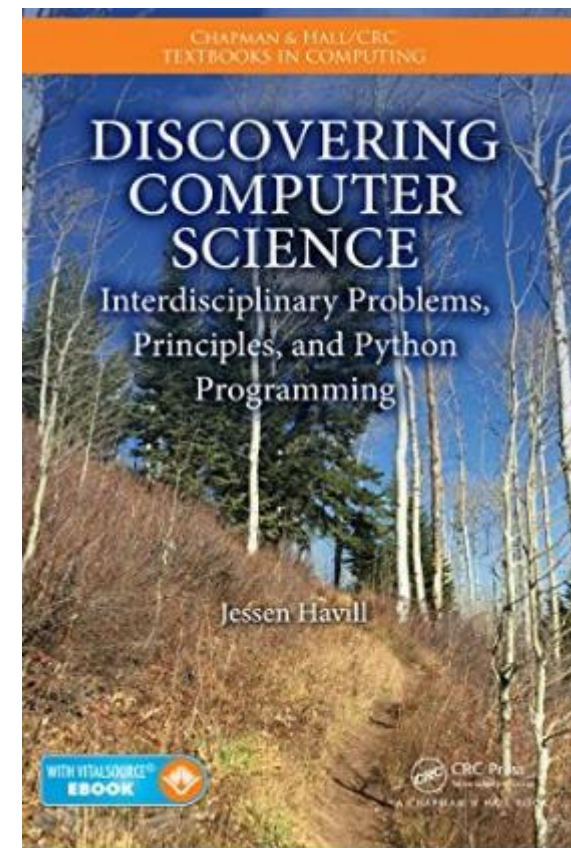
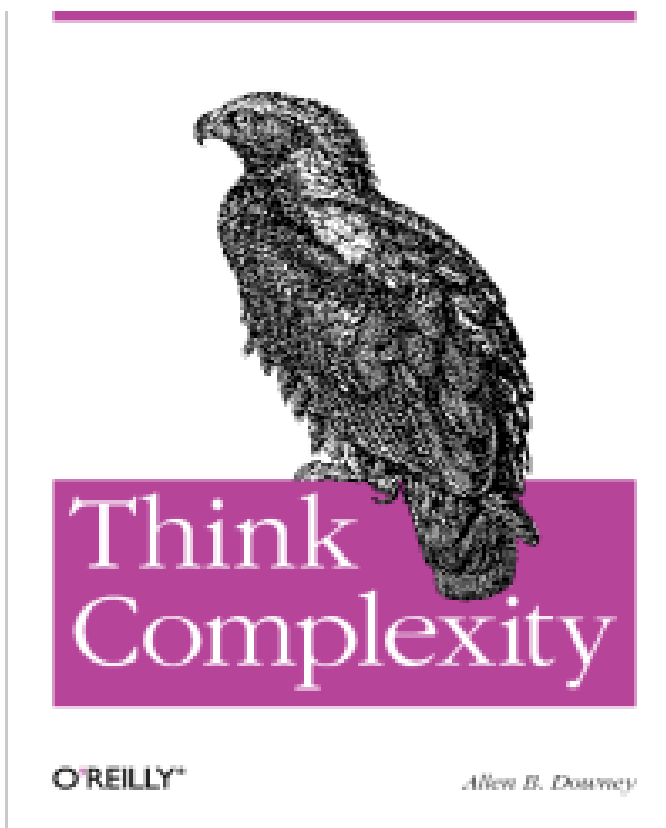
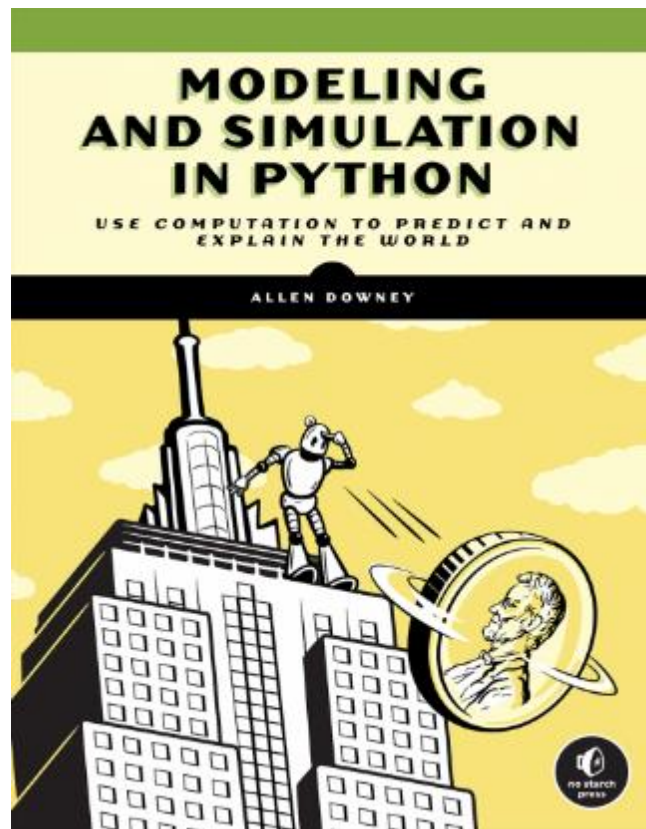
Natural Selection



Game of Life



Resources



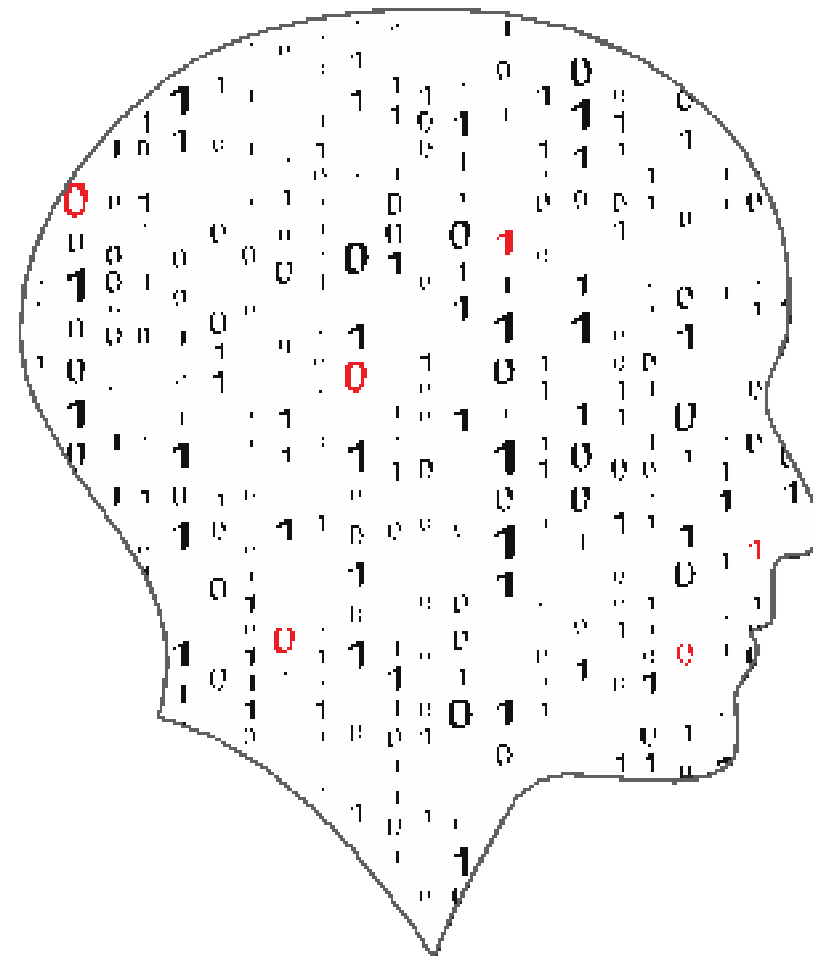


Oide

Tacú leis an bhFoghlaim
Ghairmiúil i measc Ceannairí
Scoile agus Múinteoirí

Supporting the Professional
Learning of School Leaders
and Teachers

Reflection



LEAVING CERTIFICATE
COMPUTER SCIENCE





Snowball Reflection (Wider Group Reflection)

Instructions :

- Individually reflect on the question posed (next slide) and write your thoughts on the paper provided.
- Scrunch up the page of paper and walk to the closest wall.
- On “GO” throw the piece of paper towards the centre of the room.
- Carefully, collect a “snowball” different from your own, read it, reflect upon it and discuss it with a colleague from a different table.





Snowball Reflection



How will I provide my students with opportunities
to learn more about
modelling and simulation?



Oide



An Roinn Oideachais
Department of Education



Tacú leis an bhFoghlaim
Ghairmiúil i measc Ceannairí
Scoile agus Múinteoirí

Supporting the Professional
Learning of School Leaders
and Teachers



Oide

Tacú leis an bhFoghlaim
Ghairmiúil i measc Ceannairí
Scoile agus Múinteoirí

Supporting the Professional
Learning of School Leaders
and Teachers

National Workshop 4

Session 5

ALT3: Investigate and Plan





Part 1	ALT3 and the LCCS specification
Part 2	Group activity 1: ALT3 Investigate
Part 3	Group activity 2: ALT3 Plan (including feedback)



By the end of this session participants will have:



worked in groups to share and evaluate potential ideas for ALT 3 (modelling & simulation)



collaborated on developing one potential idea for ALT 3 further



given and received feedback on potential ALT 3 ideas



enhanced their understanding of the Investigate and Plan stages of the Design Process with a particular focus on ALT 3



Key



Booklet



Individual Activity



Group Activity



Reflection



Group Feedback



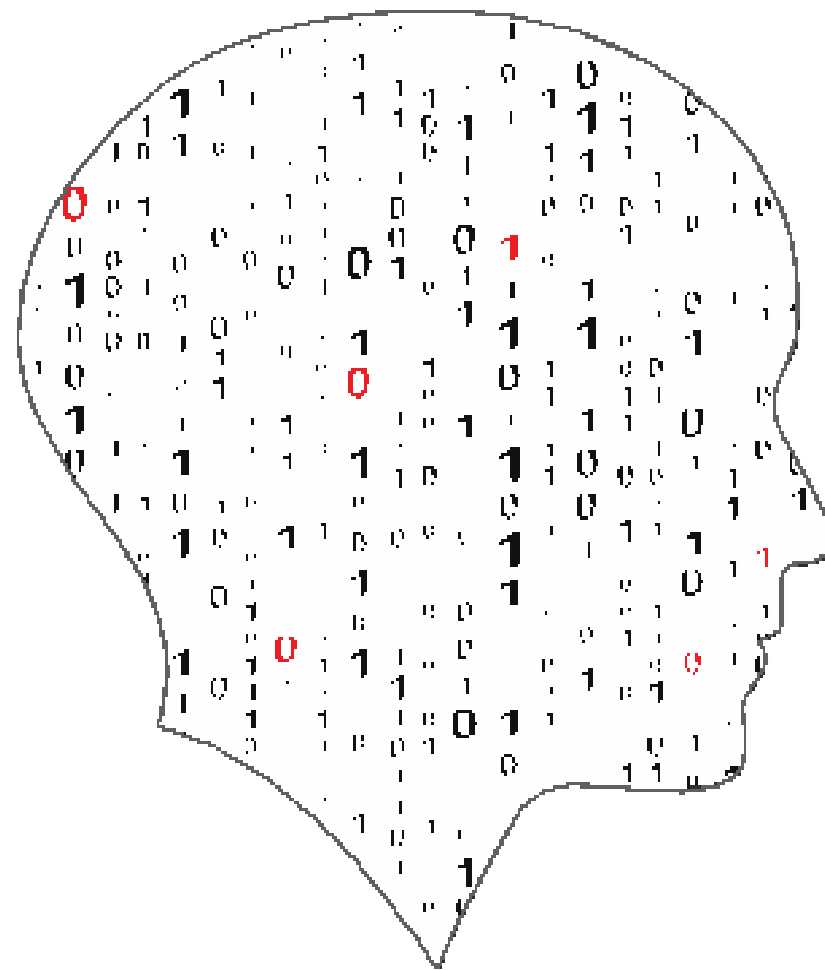
Oide

Tacú leis an bhFoghlaim
Ghairmiúil i measc Ceannairí
Scoile agus Múinteoirí

Supporting the Professional
Learning of School Leaders
and Teachers

Applied Learning Task 3 (ALT3)

Modelling and Simulation



LEAVING CERTIFICATE
COMPUTER SCIENCE





Context

Strand 1: Practices and principles	Strand 2: Core concepts	Strand 3: Computer science in practice
<ul style="list-style-type: none">▶ Computers and society▶ Computational thinking▶ Design and development	<ul style="list-style-type: none">▶ Abstraction▶ Algorithms▶ Computer systems▶ Data▶ Evaluation/Testing	<ul style="list-style-type: none">▶ Applied learning task 1<ul style="list-style-type: none">- Interactive information systems▶ Applied learning task 2 - Analytics▶ Applied learning task 3<ul style="list-style-type: none">- Modelling and simulation▶ Applied learning task 4<ul style="list-style-type: none">- Embedded systems



Applied Learning Tasks (ALTs)

‘Students work in teams to carry out four applied learning tasks over the duration of the course.’

‘Each of which results in the creation of a real or virtual computational artefact and a report.’

‘Where possible, the artefacts should be beneficial to the community and society in general.’



‘These artefacts should relate to the students’ lives and interests.’

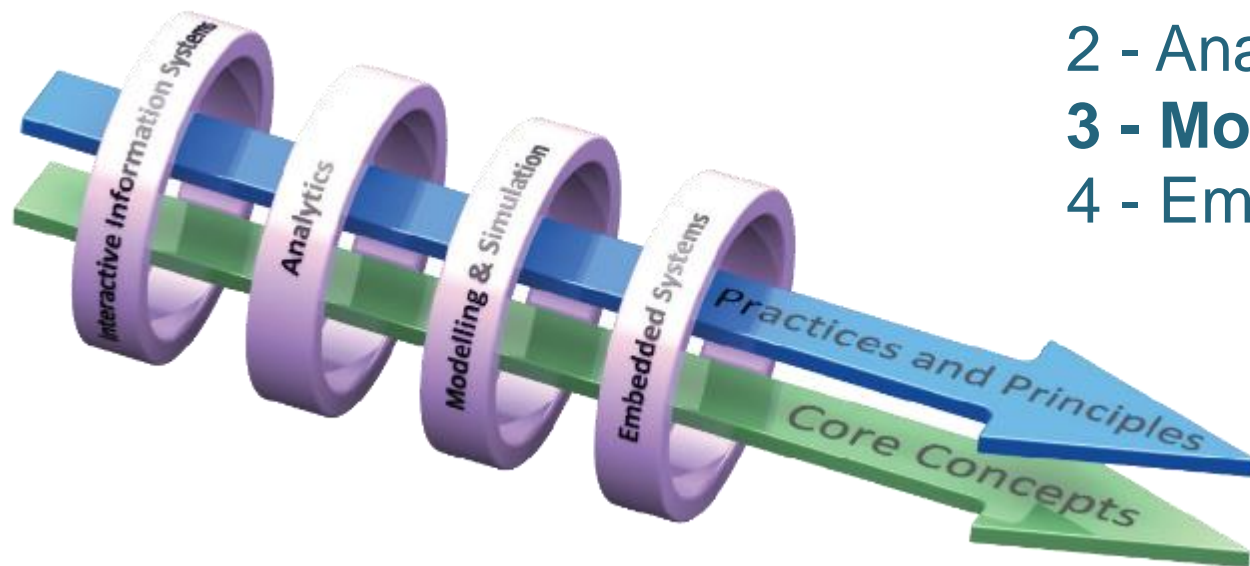
‘Examples of computational artefacts include programs, games, web pages, simulations, visualisations, digital animations, robotic systems, and apps.’



LCCS Interwoven

The four applied learning tasks explore the four following contexts:

- 1 - Interactive information systems
- 2 - Analytics
- 3 - **Modelling and simulation**
- 4 - Embedded systems



Key point to remember: Explore and teach the LOs through the lens of ALTs.



ALT 3: Modelling and Simulation

“Modelling, programming and coding require careful analysis of patterns and relationships to solve problems.”

“Students will engage with a problem that is difficult to solve analytically, but that is amenable to a solution using simulation or modelling.”

“Students will develop a computer system that simulates or models the problem. Engaging with a problem in this way will heighten students’ awareness and understanding of how algorithms can be used across a wide range of disciplines and subjects.”



ALT 3: Learning outcomes

- 3.8 develop a model that will allow different scenarios to be tested
- 3.9 analyse and interpret the outcome of simulations **both before and after modifications have been made**
- 3.10 **explain the benefits of using agent-based modelling and how it can be used to demonstrate emergent behaviours**



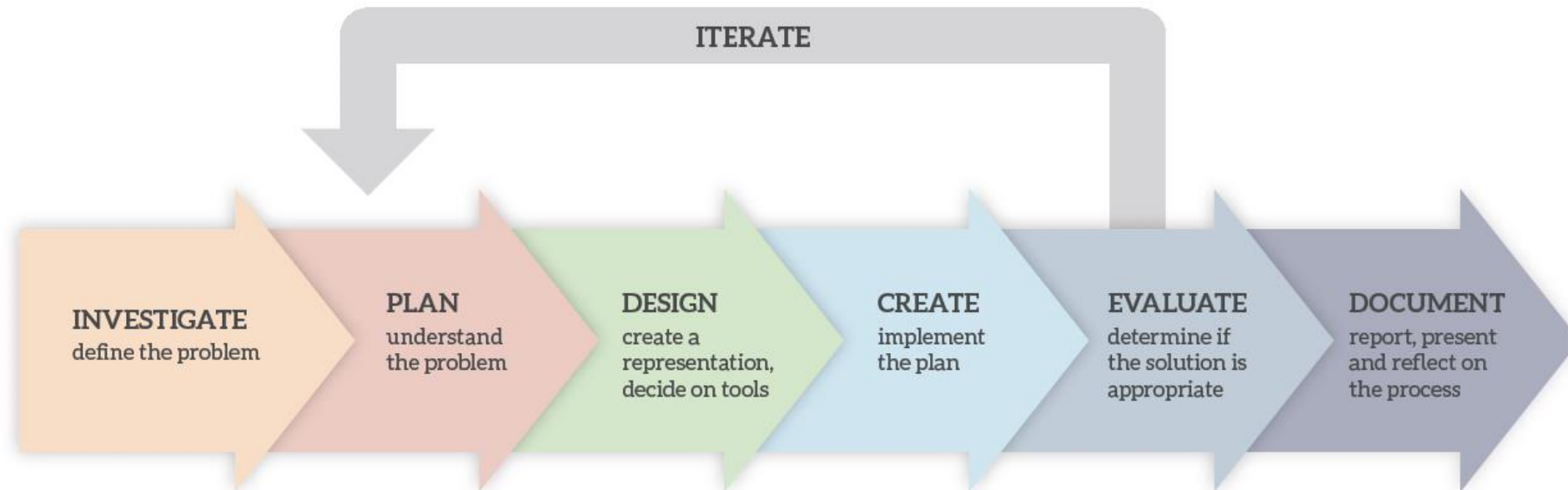
Key Message to remember:

Explore and teach the LOs through the lens of ALTs.

There are numerous ways to achieve this.

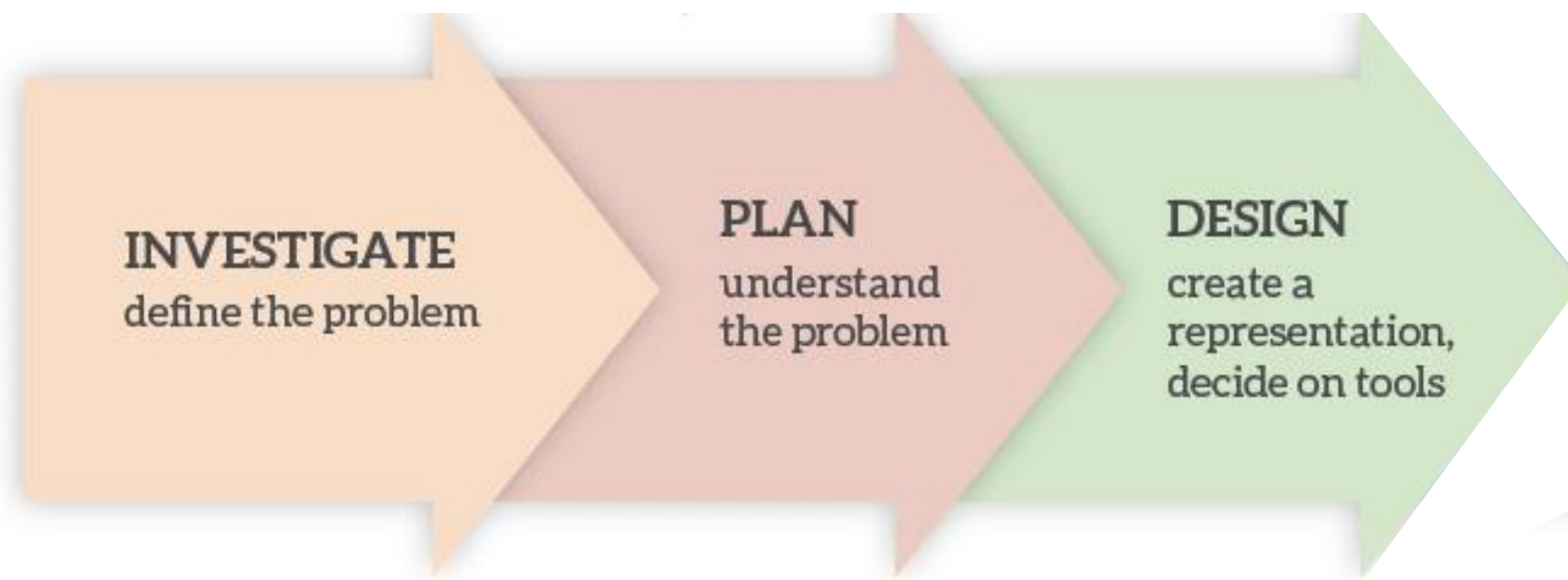


The Design Process





The Design Process





ALT3 Activity: Investigate



In your groups, share **potential ideas** for ALT 3.

Aim for **as many ideas** as you can.

Add your ideas to the **Google doc**.

Nominate a timekeeper, notetaker, chairperson and spokesperson.



P. 30



15 mins



Features/characteristics of models

“All models are wrong but some are useful” – George Box

Forecasting/predicting (future)

Messy

Real-life

Not easily solved by other means



ALT 3 Activity: Plan

In your assigned groups, **evaluate** your potential ideas for ALT 3.

Choose **one idea** for further development.

Develop a **functional specification** for your chosen idea.

You may use the **prompt questions** on next slide/shared document to help you.



P. 31



20 mins

ALT3 Activity: Plan – prompt questions



Choose one idea and consider the following prompts:

- What will your project do/not do?
- Aims? Any limitations?
- Who are the end users?
- What are the group roles and responsibilities?
- What are the tools/materials required?
- Will your project allow you to make predictions/decisions more easily?
- What is new and has to be researched further?
- What other LOs can be experienced through the lens of this project?
- Are there any ethical issues?



P. 32



20 mins



Feedback





Oide



An Roinn Oideachais
Department of Education





Oide

Tacú leis an bhFoghlaim
Ghairmiúil i measc Ceannairí
Scoile agus Múinteoirí

Supporting the Professional
Learning of School Leaders
and Teachers

National Workshop 4

Session 6

ALT3: Design and Create



Session Overview



Section 1	ALT Design Process - Reporting
Section 2	ALT3: Design/Create and Feedback
Section 3	Resources to support ALT 3 (Modelling and Simulation)

By the end of the day participants will have had the opportunity to:



Oide



enhance their understanding of the reporting requirements with regard to the coursework assessment (CWA)



work in groups to **design one** potential solution for ALT 3



given and received feedback on potential ALT 3 designs



explore a variety of models from online resources and Python code provided



Key



Booklet



Individual Activity



Group Activity



Reflection



Group Feedback



Take Notes



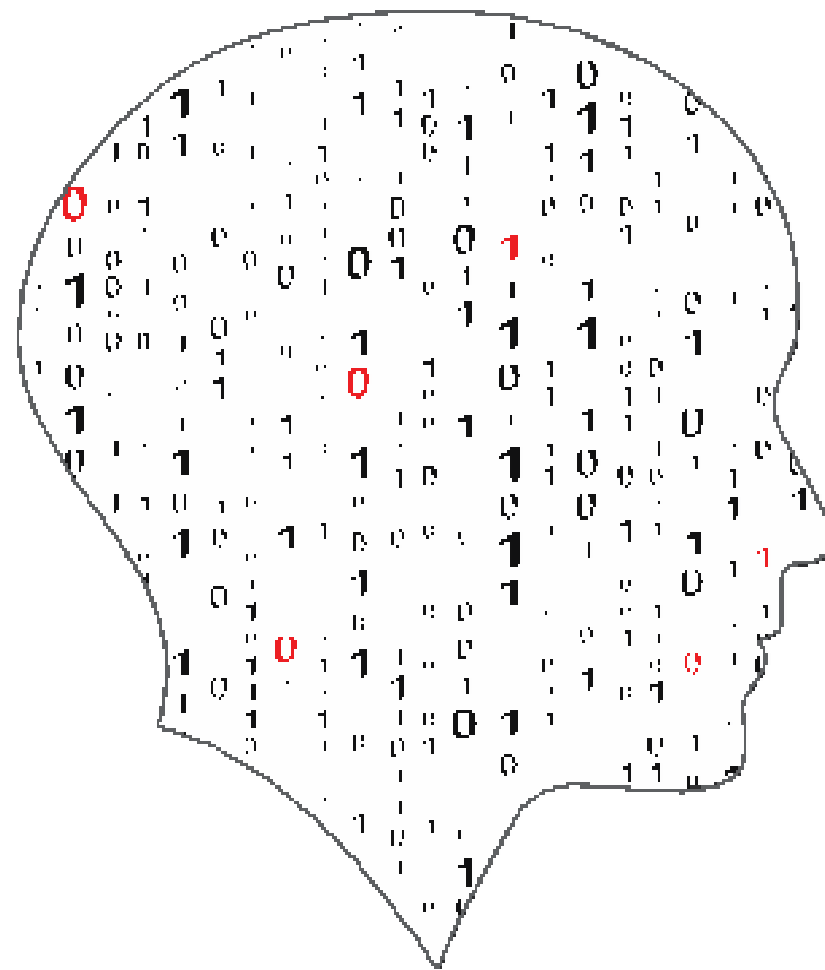
Oide

Tacú leis an bhFoghlaim
Ghairmiúil i measc Ceannairí
Scoile agus Múinteoirí

Supporting the Professional
Learning of School Leaders
and Teachers

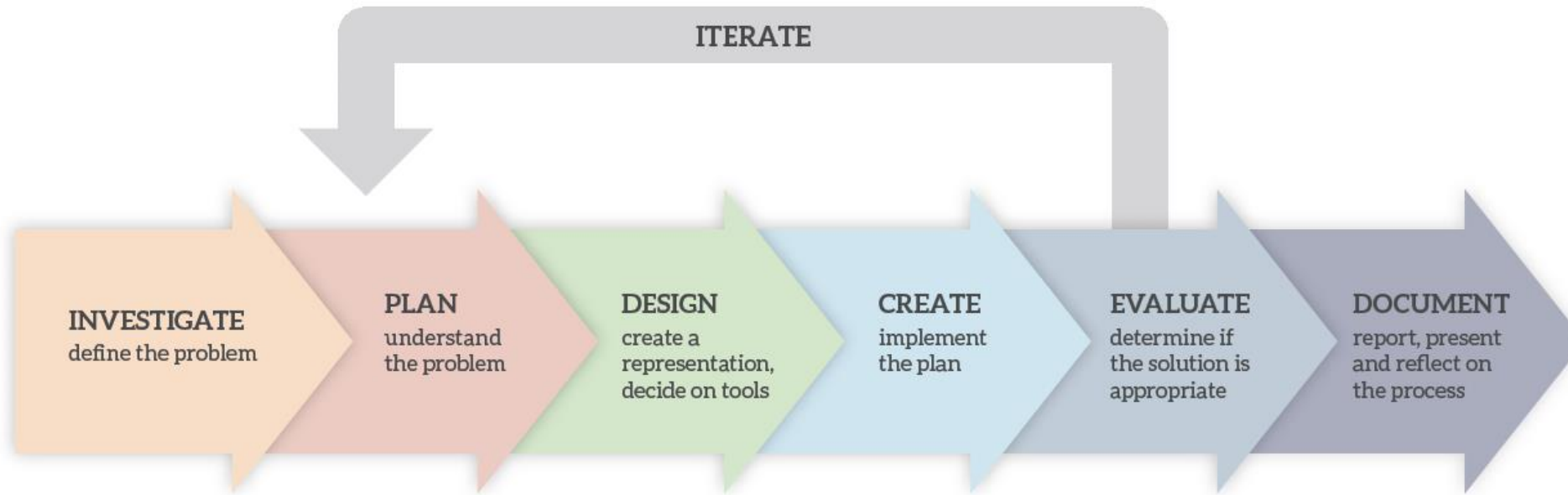
Section I

ALT Design Process - Reporting



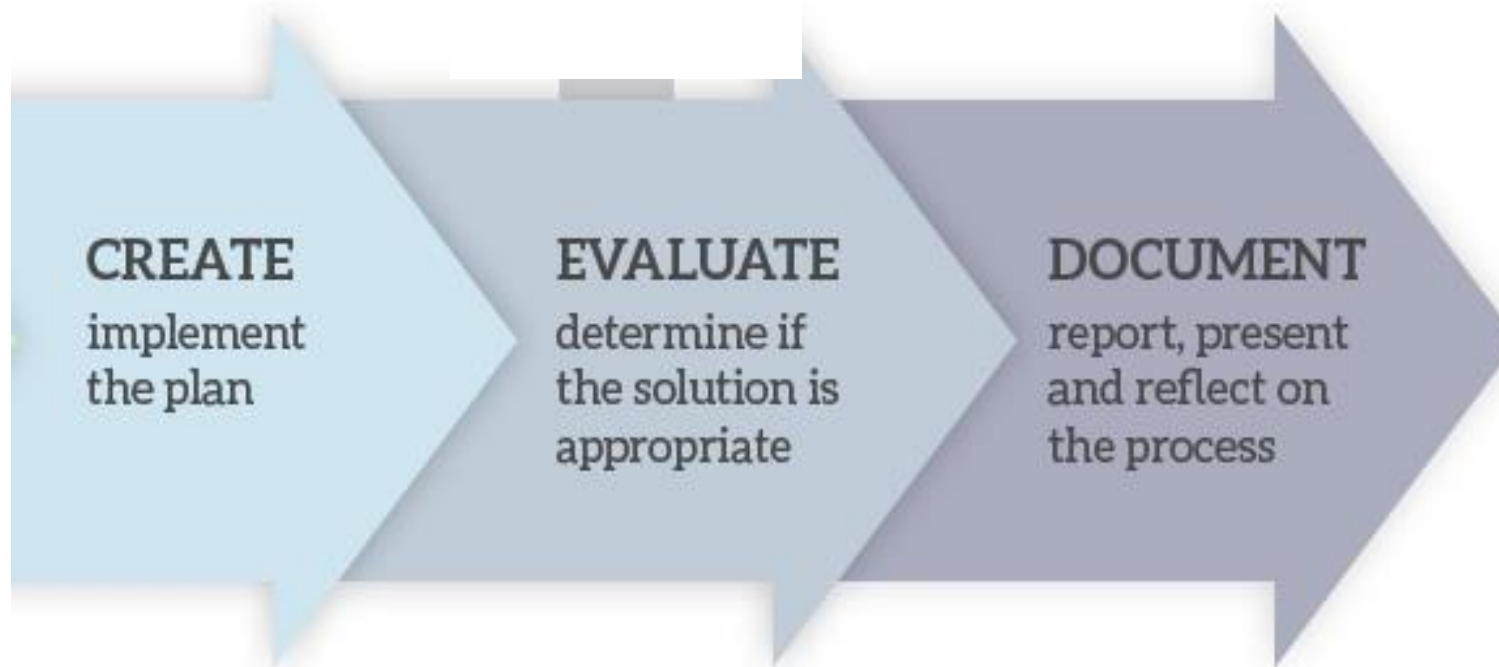
LEAVING CERTIFICATE
COMPUTER SCIENCE





LCCS Specification: p11

The Design Process





Reporting - ALTs



The output from each task is a computational artefact and a concise individual report outlining its development. In the report, students outline where and how the core concepts were employed. The structure of the reports should reflect the design process shown above in Figure 3. Initial reports could be in the form of structured presentations to the whole class. As students progress, reports should become detailed and individual. Reports are collected in a digital portfolio along with the computational artefact and must be verified as completed by both the teacher and the student. The (separate) externally-assessed coursework will be based on all learning outcomes, with those of strand 3 being particularly relevant.

LCCS Specification: p19

They (students) will develop skills in communication by collaborating to generate reports and present them to their peers. The strand 3 tasks will enable students to take an active role in their own learning by setting goals, developing action plans, and receiving and responding to assessment feedback.

LCCS Specification: p15

S1: Designing and developing	
Design process	1.19 identify features of both staged and iterative design and development processes
Working in a team, assigning roles and responsibilities	1.20 collaborate and assign roles and responsibilities within a team to tackle a computing task
Communication and reporting	1.21 identify alternative perspectives, considering different disciplines, stakeholders and end users 1.22 read, write, test, and modify computer programs
Software development and management	1.23 reflect and communicate on the design and development process



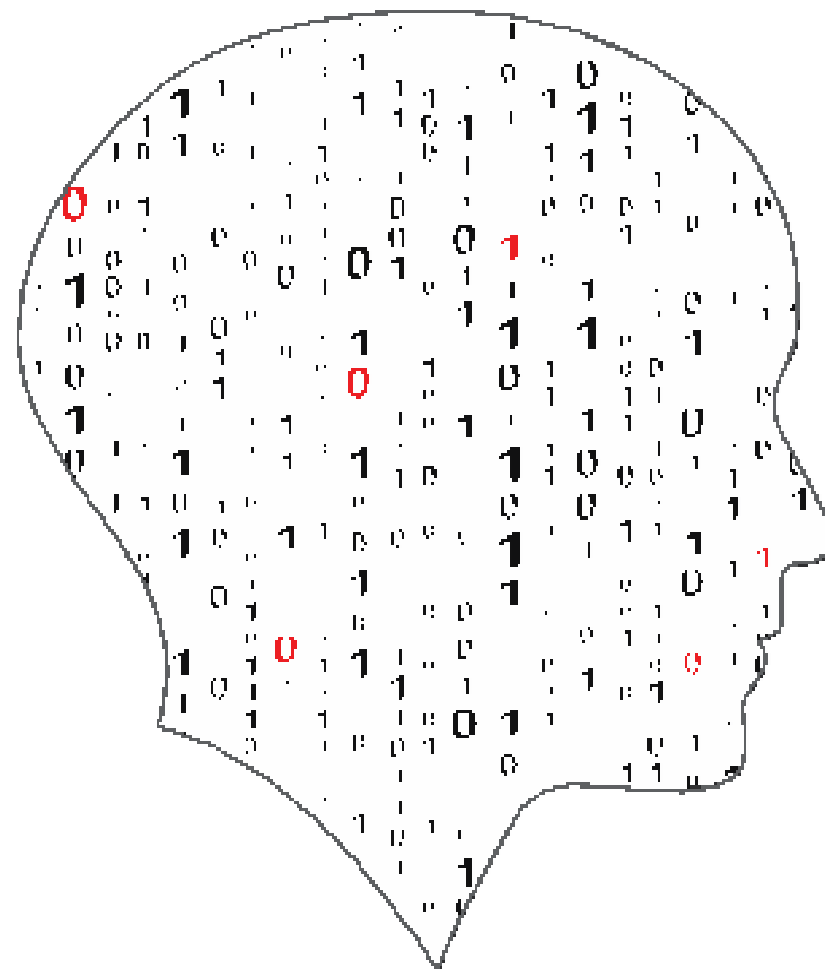
Oide

Tacú leis an bhFoghlaim
Ghairmiúil i measc Ceannairí
Scoile agus Múinteoirí

Supporting the Professional
Learning of School Leaders
and Teachers

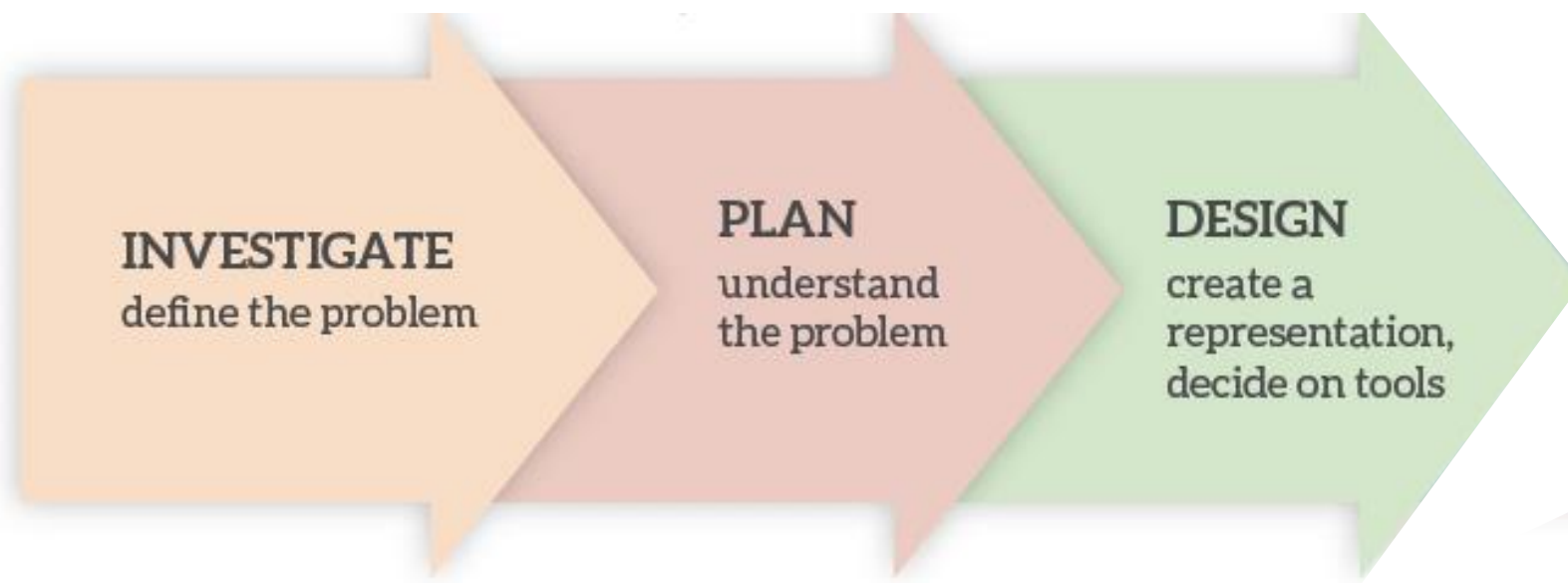
Section II

ALT3: Design/Create
and Feedback

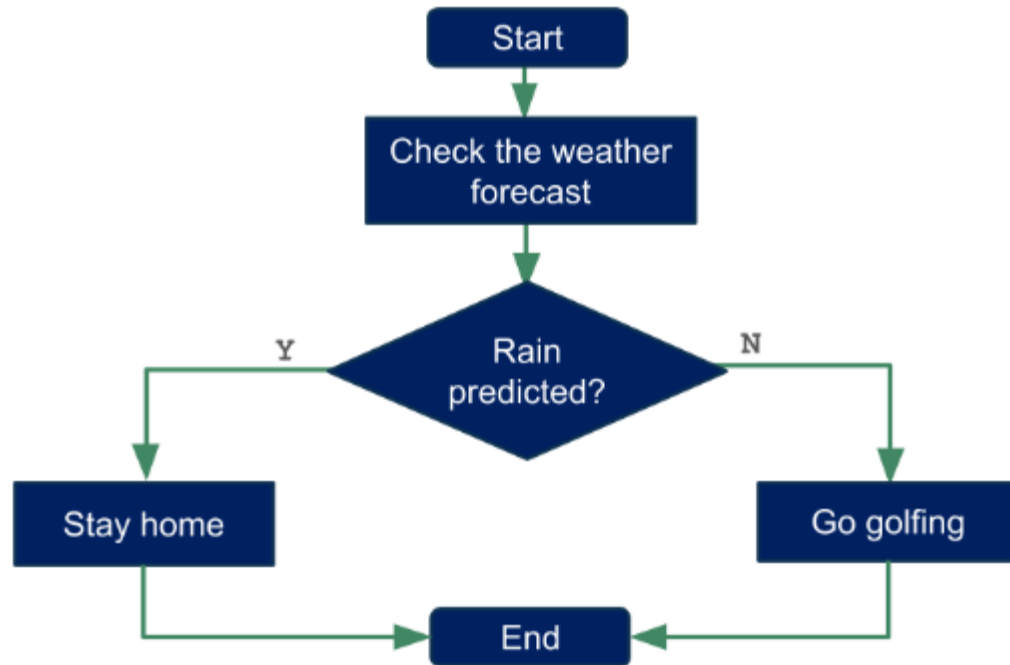


LEAVING CERTIFICATE
COMPUTER SCIENCE



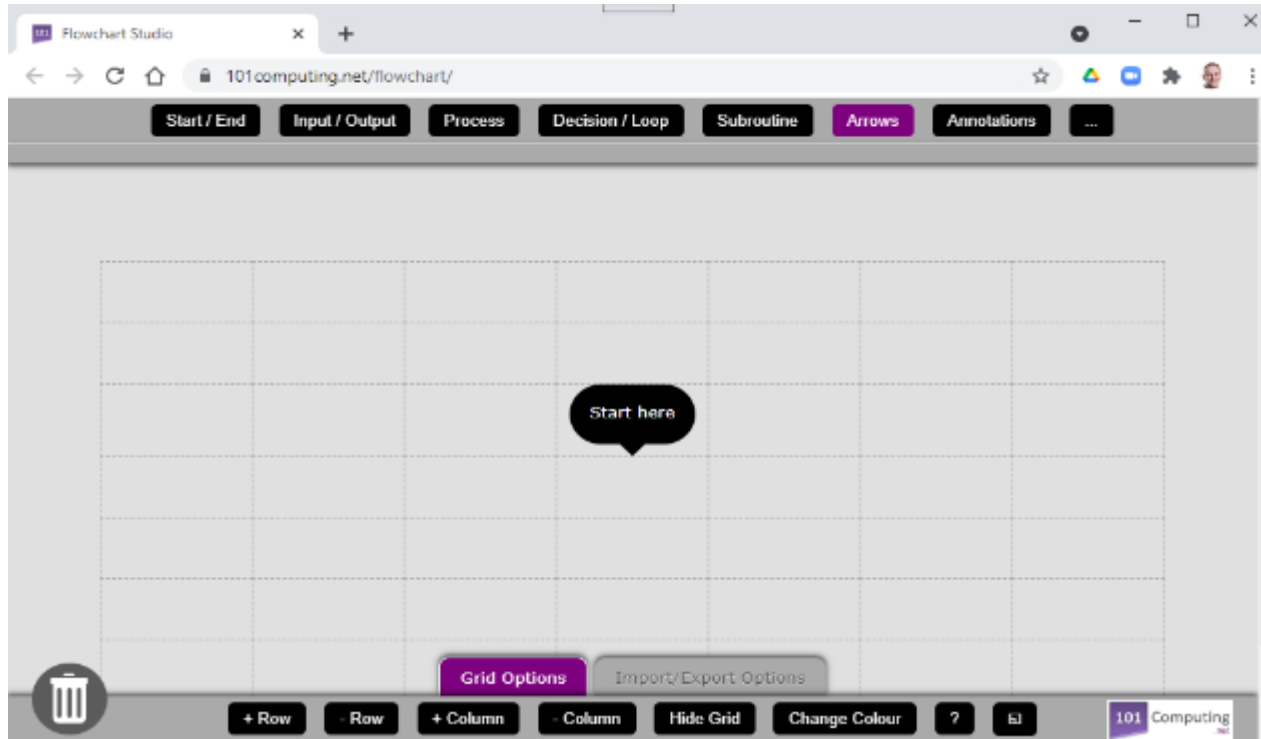


Should I play golf?





Online flowcharting



Should I play golf?



P. 32

<https://www.101computing.net/flowchart/>



Group Activity: ALT3 Design



P. 34



20 mins



3

DESIGN
create a
representation,
decide on tools

ITERATE

Input: The requirements document (the plan)

- How is the system realised?
- What is the overall design?
- What are the components of the system (system architecture)? Are there sub-systems (e.g. functions)? What are they?
- How is the data represented (variables, data types)?
- How is the data captured? What about data validation? (Data flow diagrams)
- What algorithms are needed? (Flow charts, pseudocode)
- What about test cases?
- What are the milestones and timelines?

Output: A representation of the system (system architecture diagrams, data flow diagrams, flow charts, pseudocode, etc).



P. 32



20 mins



Presentation & Debrief

What steps?

Why?

Any areas of disagreement?

Did you consider alternatives?

How?

System architecture?

Algorithms?

How could you improve?

What is your design?





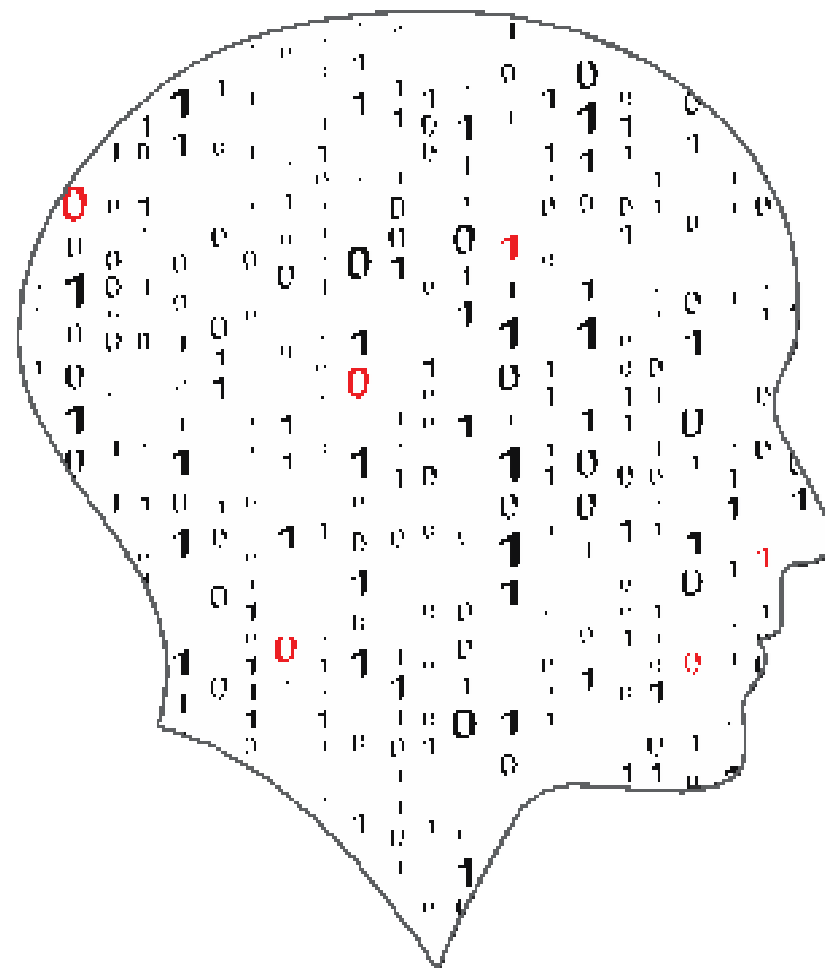
Oide

Tacú leis an bhFoghlaim
Ghairmiúil i measc Ceannairí
Scoile agus Múinteoirí

Supporting the Professional
Learning of School Leaders
and Teachers

Section III

Resources to support
ALT 3 (Modelling and
Simulation)



LEAVING CERTIFICATE
COMPUTER SCIENCE

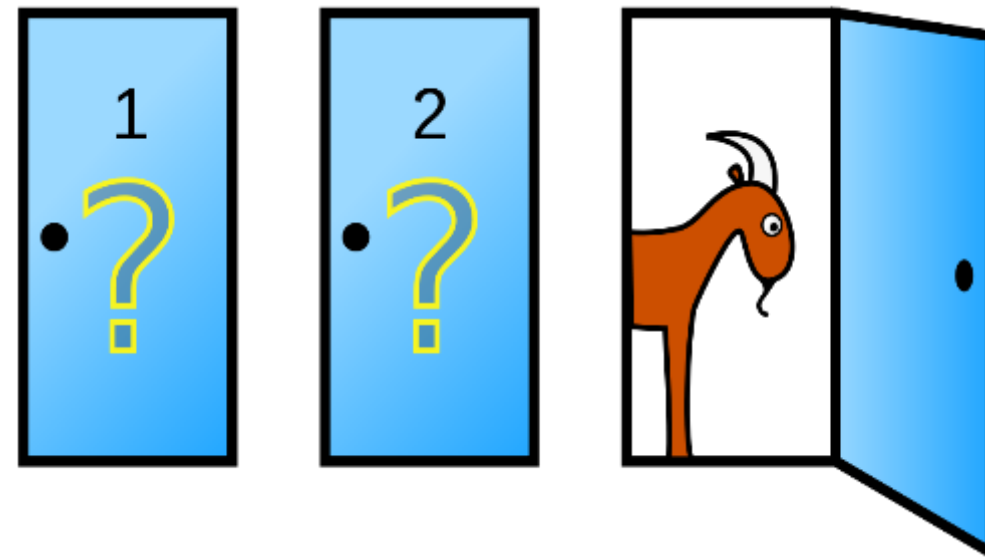




Resources and Examples



NCCA Population Example



Monty Hall Example



Models to explore post-workshop (next slide)

1. Some questions to consider when looking at these models are:
2. Is this an Agent Based model?
3. What is being modelled?
4. What are the agents?
5. Is there any emergent behaviour?
6. In what ways are models like this different to the fish population model?





Online Resources



Netlogo



Project GUTS



PhET



Pixar



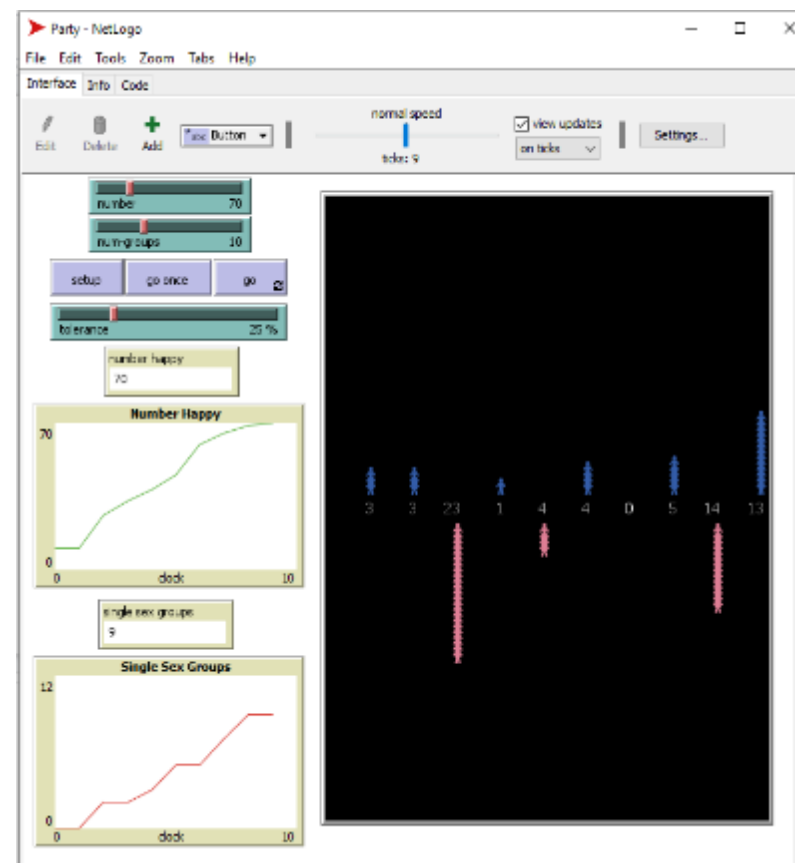
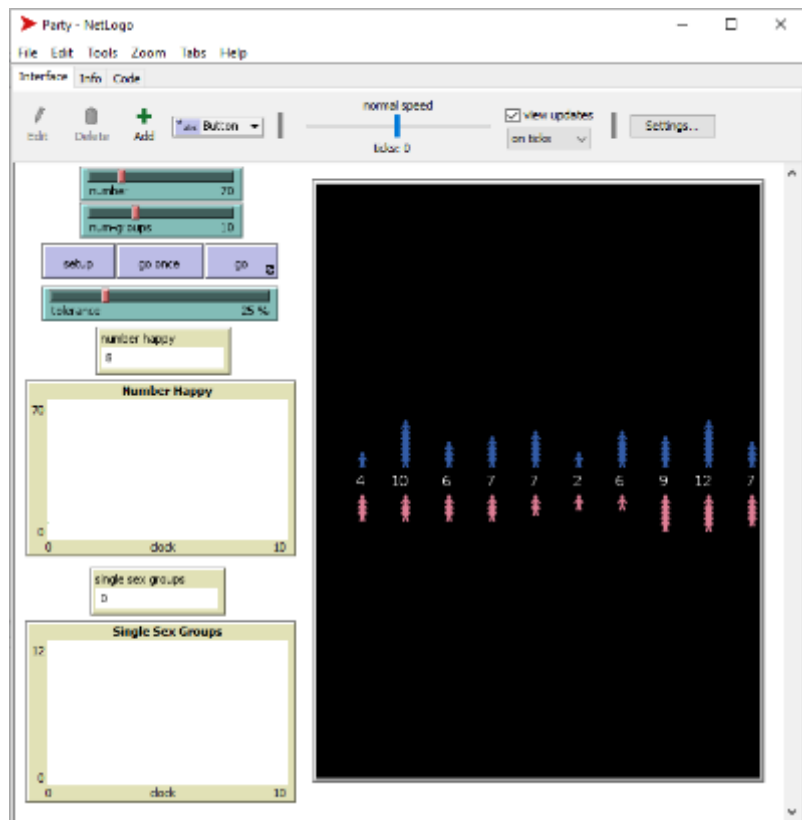
Natural Selection



Game of Life

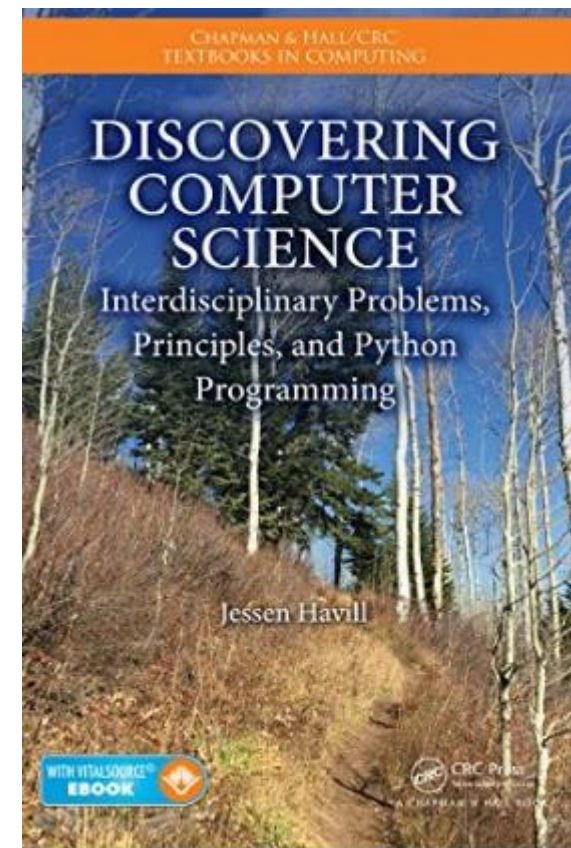
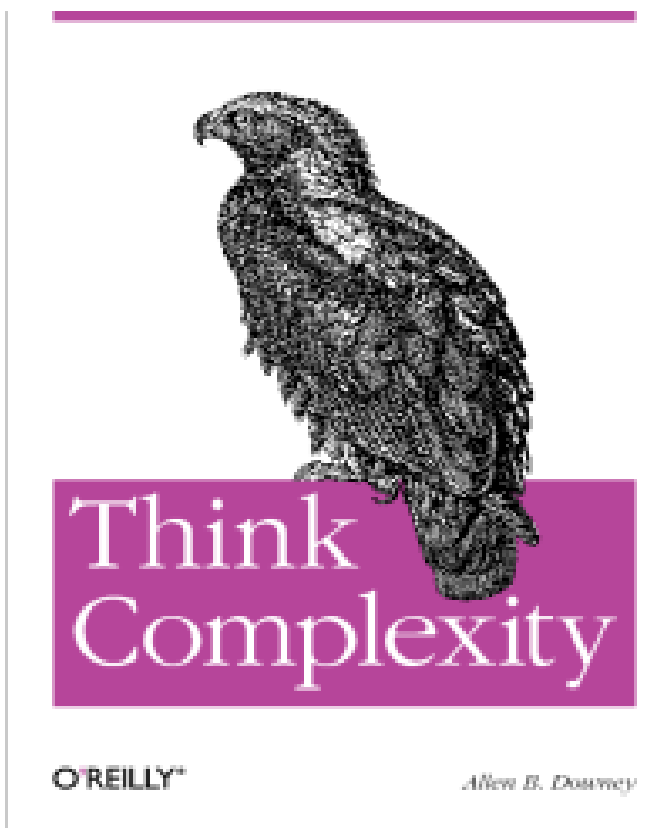
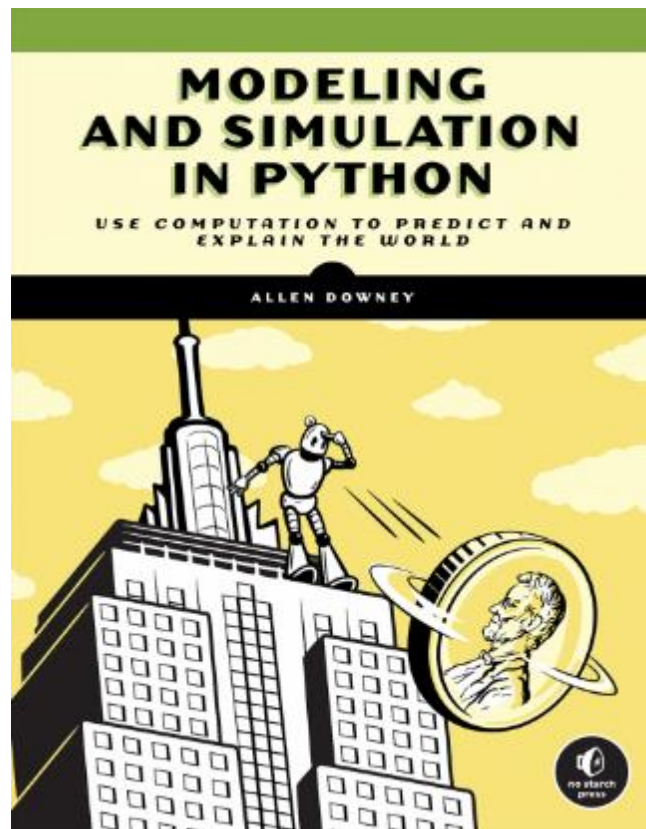


NetLogo Example





Resources





Oide



An Roinn Oideachais
Department of Education

