PDST
Professional Development Service for Teachers
An tSeirbhís um Fhorbairt Ghairmiúil do Mhúinteoirí

An Roinn Oideachais agus Scileanna
Department of Education and Skills

National Workshop 1

LEAVING CERTIFICATE
COMPUTER SCIENCE

PDST
Professional Development Service for Teachers | An tSeirbhís um Fhorbairt Ghairmiúil do Mhúinteoirí

Computational Thinking

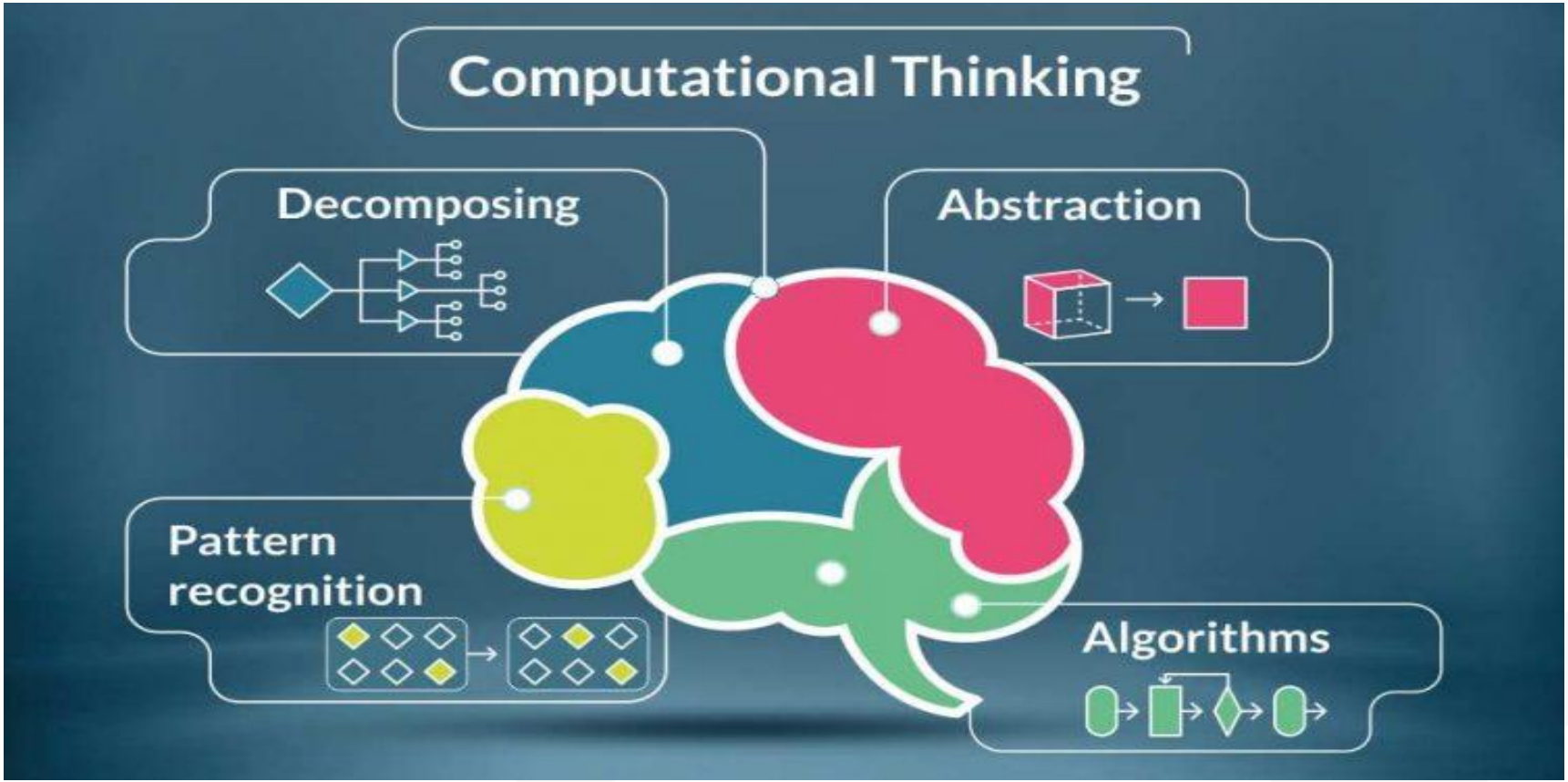
What is Computational Thinking?

- Problem Solving Methods: for a ‘Computer’
- Help

- Several Models: eg 4 Concepts / Pillars:
- *Decomposition*
- *Pattern Recognition*
- *Abstraction*
- *Algorithm Formation*

- *Different subjects… even English Literature*

Looking at the Monty Hall Problem


Computational Thinking
Decomposing
Abstraction
Pattern recognition
Algorithms

Computational Thinking

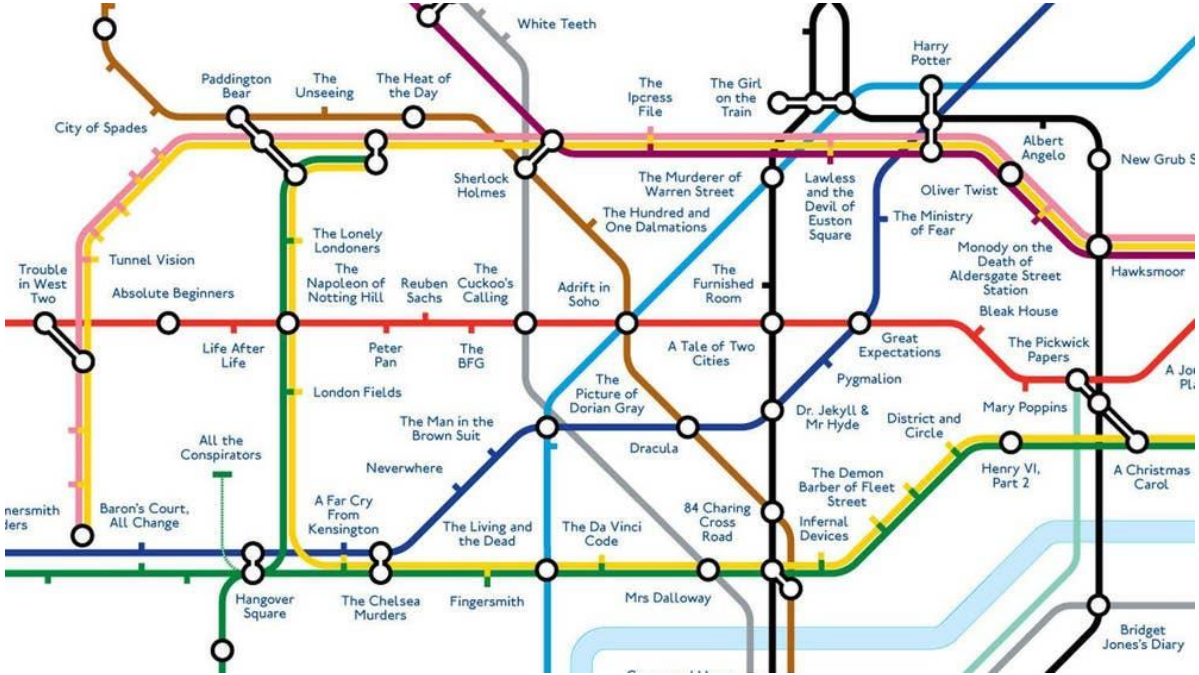
- Monty Hall Problem
- How to Subtract
- London Underground
- Money – Change (Activity): White board
- Sisters

Computational Thinking

234-159

1.Modern Primary School way.
2.Old Primary school way.
3.Shop Assistant.
4.Dart player.

The Tube



How is abstraction used? – Menti **8240 9970**

CT Activity 1

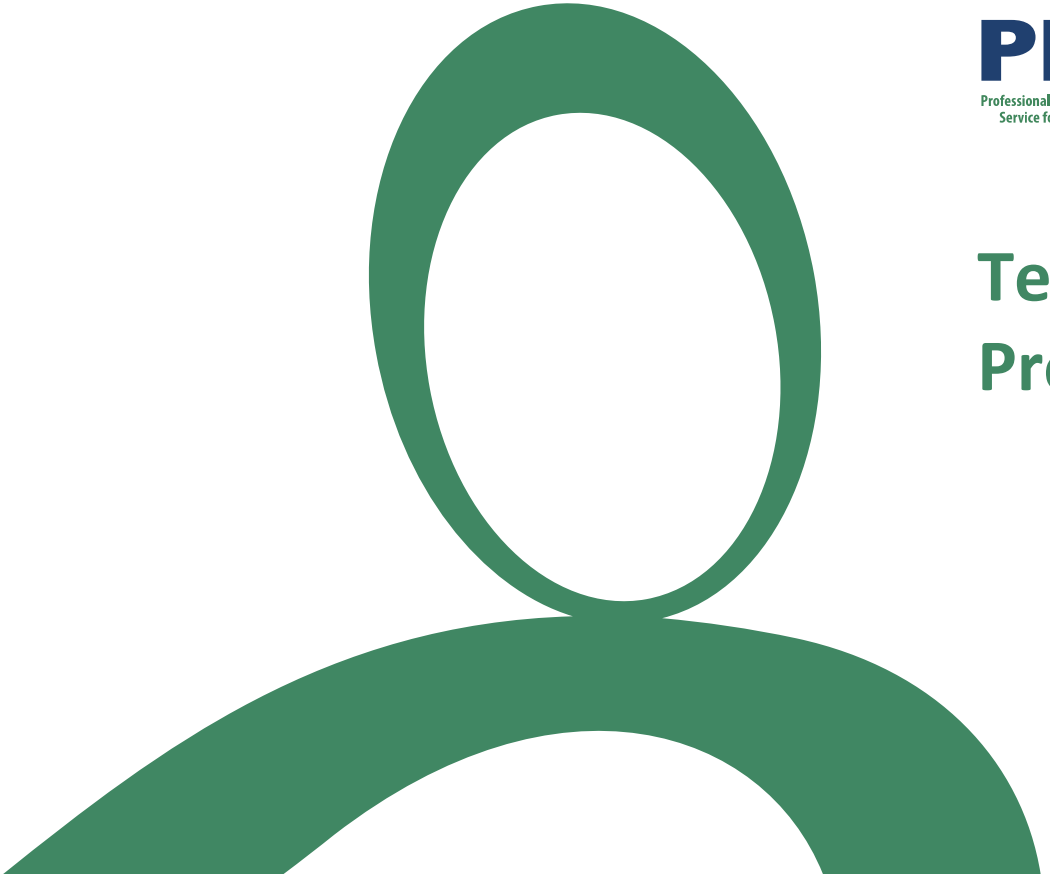
Euro coins are issued in the denominations shown

€1 = 100c

What is the minimum number of coins required to make up €27.93 cents?

CT – Activity 2

Who has more sisters, boys or girls?

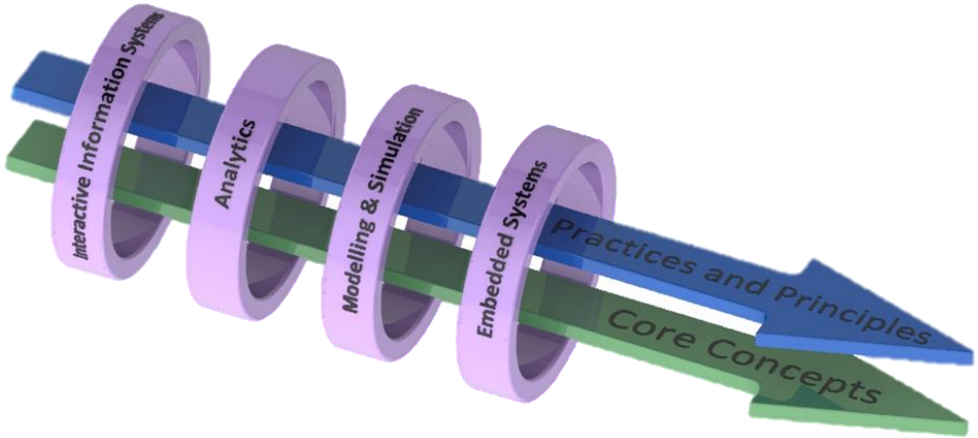


PDST
Professional Development Service for Teachers | An tSeirbhís um Fhorbairt Ghairmiúil do Mhúinteoirí

Teaching and Learning Programming

Programming and LCCS

"The role of programming in computer science is like that of practical work in the other subjects—it provides motivation, and a context within which ideas are brought to life. Students learn programming by solving problems through computational thinking processes and through practical applications such as applied learning tasks." LCCS specification (2017)



The four applied learning tasks explore the four following contexts:

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

A Grand Challenge!

"Teaching and learning programming is considered one of the grand challenges of computing education"

Casperen, 2018

"Programming is a hard craft to master and its teaching is challenging"

Crick, 2017

"Programming is not an easy subject to be studied"

Lahtinen et. al., 2005

"Teaching students to program is a complex process"

Luxton-Reilly et. al., 2018

How did you learn how to program?

What were main challenges for you?

Learning Challenges faced by Novice Programmers

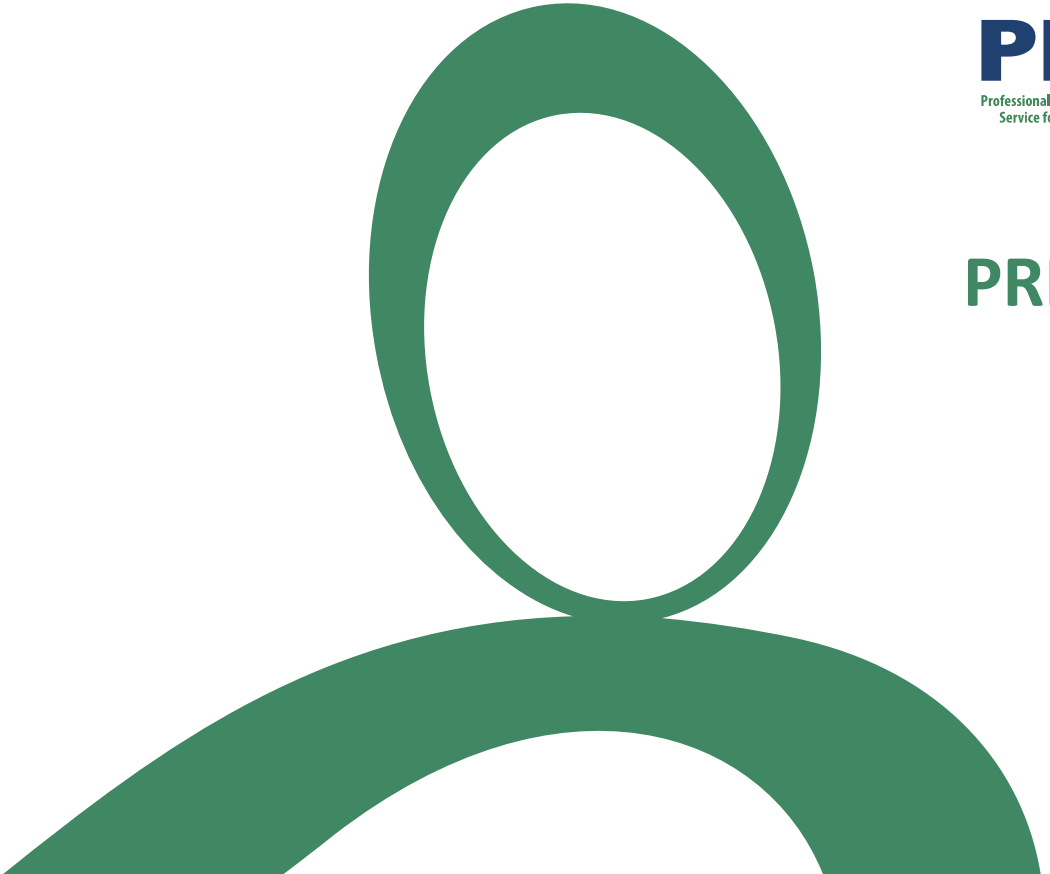
Programming IS difficult to learn because it requires and understanding of:

- the underlying machine that one is trying to control (Notational Machine)
- the problem to be solved (Orientation)
- the language syntax and semantics (Notation / Vocabulary)
- how to apply the language to solve problems (Application / Design)
- the development environment (Pragmatics and Perspectives)
- different levels of abstraction

Cognitive Load

What research tells us about learning programming

- Read before you write – there is clear evidence that tracing and explaining code helps (Lister, numerous papers)
- Levels of abstraction - problem at higher-level; syntax at lower-level (Perrenet & Kassenbrod, 2006)
- Threshold concepts in programming cause difficulties for students (Sanders & McCartney, 2016)
- Misconceptions abound (research from 1980s to current day – Juha Sorva)
- Growth mindset – popular approach, with some research evidence in Computing (Cutts, 2010)
- Programming Pedagogies help – dialogue and discussion at a logical level help students (Porter et al, 2012)



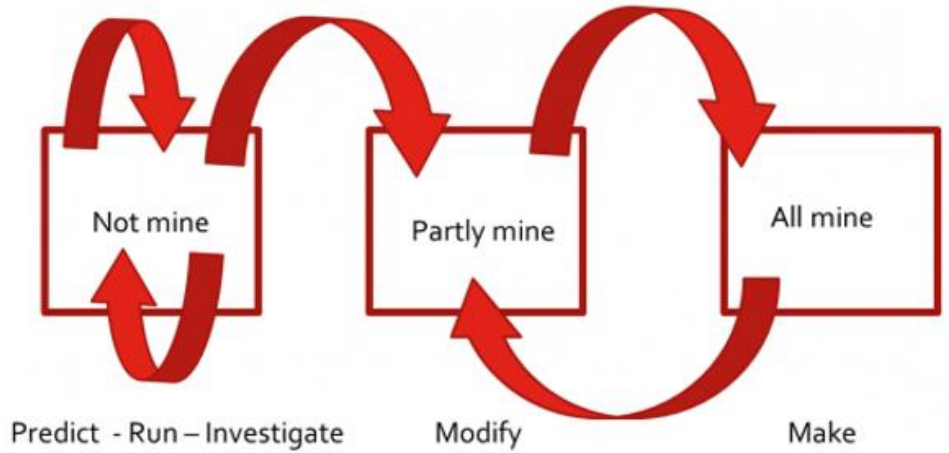
PDST
Professional Development Service for Teachers | An tSeirbhís um Fhorbairt Ghairmiúil do Mhúinteoirí

PRIMM

PRIMM

A way of structuring programming lessons that focuses on:

- Reading before Writing
- Student Collaboration
- Reducing Cognitive Load
- Well-chosen starter programs
- Ownership Transfer



Sources:

1. https://blogs.kcl.ac.uk/cser/2017/02/20/exploring-pedagogies-for-teaching-programming-in-school/ (Sue Sentence)
2. https://blogs.kcl.ac.uk/cser/2017/09/01/primm-a-structured-approach-to-teaching-programming/ (Sue Sentence)
3. Sue Sentance, Jane Waite & Maria Kallia (2019) Teaching computer programming with PRIMM: a sociocultural perspective, Computer Science Education, 29:2-3, 136-176, DOI: 10.1080/08993408.2019.1608781

PRIMM

- **Predict:** given a working program, what do you think it will do? (at a high level of abstraction)
- **Run:** run it and test your prediction
- **Investigate:** What does each line of code mean? (get into the nitty gritty - low level of abstraction - trace/annotate/explain/talk about parts)
- **Modify:** edit the program to make it do different things (high and low levels of abstraction)
- **Make:** design a new program that uses the same nitty gritty but that solves a new problem.

PRIMM – Example (1 of 2)

```
1. import random
2.
3. number = random.randint(1, 10)
4. #print(number)
5.
6. guess = int(input("Enter a number between 1 and 10: "))
7.
8. if guess == number:
9.     print("Your guess was correct")
10.    print("Goodbye")
11.else:
12.    print("Incorrect guess")
13.    print("Goodbye")
```

Predict: Discuss in pairs. What do you think the above program will do? Be precise. Be succinct.

Run: Download the program / Key it in. Execute the program. Test your prediction. Were you correct?

Things to think about. How to …

Investigate: Devise some questions to elicit student learning and curiosity. What if … Try … Explain …

Modify: Suggest some simple extensions / modifications for students to make in pairs. Same program.

Make: Formulate new problems that are conceptually similar. New context. New program (copy+paste)

PRIMM – Example (2 of 2)

```
1. import random
2.
3. number = random.randint(1, 10)
4. #print(number)
5.
6. guess = int(input("Enter a number between 1 and 10: "))
7.
8. if guess == number:
9.     print("Your guess was correct")
10.    print("Goodbye")
11.else:
12.    print("Incorrect guess")
13.    print("Goodbye")
```

Investigate:

1. Uncomment line 4. What happens?
2. What is the purpose of line 4?
3. What would happen if you removed `int` from line 6?
4. Try changing `==` to `!=` on line 8. What happens?
5. What if `==` was changed to `=` ?
6. What would happen if you don't enter an integer?
7. Try removing a bracket (anywhere). What happens?
8. Annotate each line of the program.

Modify:

1. Change the program so that it generates a number between 1 and 100? Can you be sure?
2. Change the program so that there is only one `print("Goodbye")` statement (without altering the logic)
3. Extend the program so that it tells the user if the number entered was *too high* or *too low*
4. Design an algorithm based on the program that would give the user 3 guesses
5. Get the computer to generate 4 numbers (lotto) OR ask the user how many numbers to generate?

Make:

1. Write a program that generates two numbers and prompts the user to enter their product

PDST
Professional Development Service for Teachers | An tSeirbhís um Fhorbairt Ghairmiúil do Mhúinteoirí

Conclusions

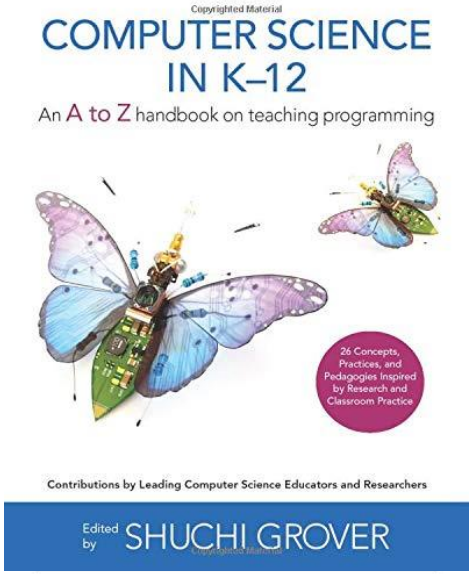
Conclusions

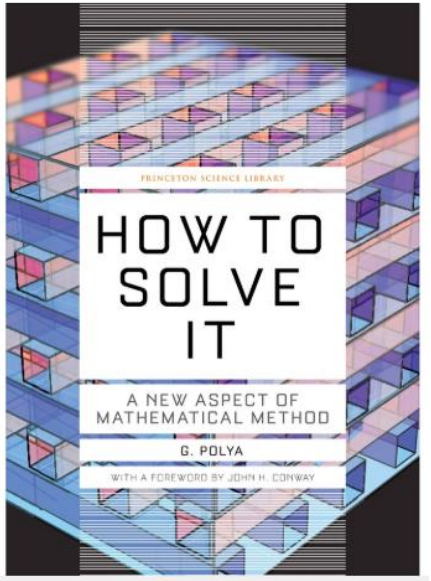
1. Programming IS difficult (for students to learn and teachers to teach)
2. Pedagogies are proven to work
3. Read/Trace code before you write
4. Constructivist approach is important
5. Growth mindset is *at least* as important as natural ability
6. Student-centric approach (teachers adopt a guide-on-the-side rather than a sage-on-the-stage approach)

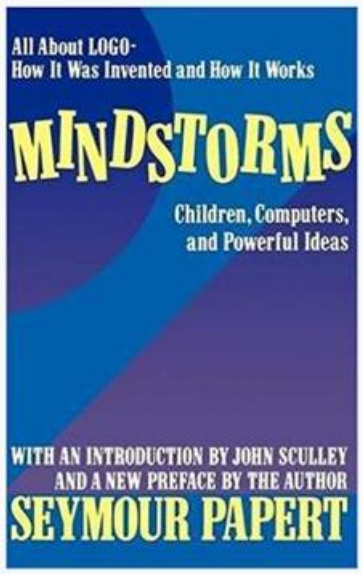
"The teacher should help, but not too much and not too little, so that the student shall have a reasonable share of the work" and, *"If the student is not able to do much, the teacher should leave him at least with some illusion of independent work."*

George Polya, How To Solve It

Resources

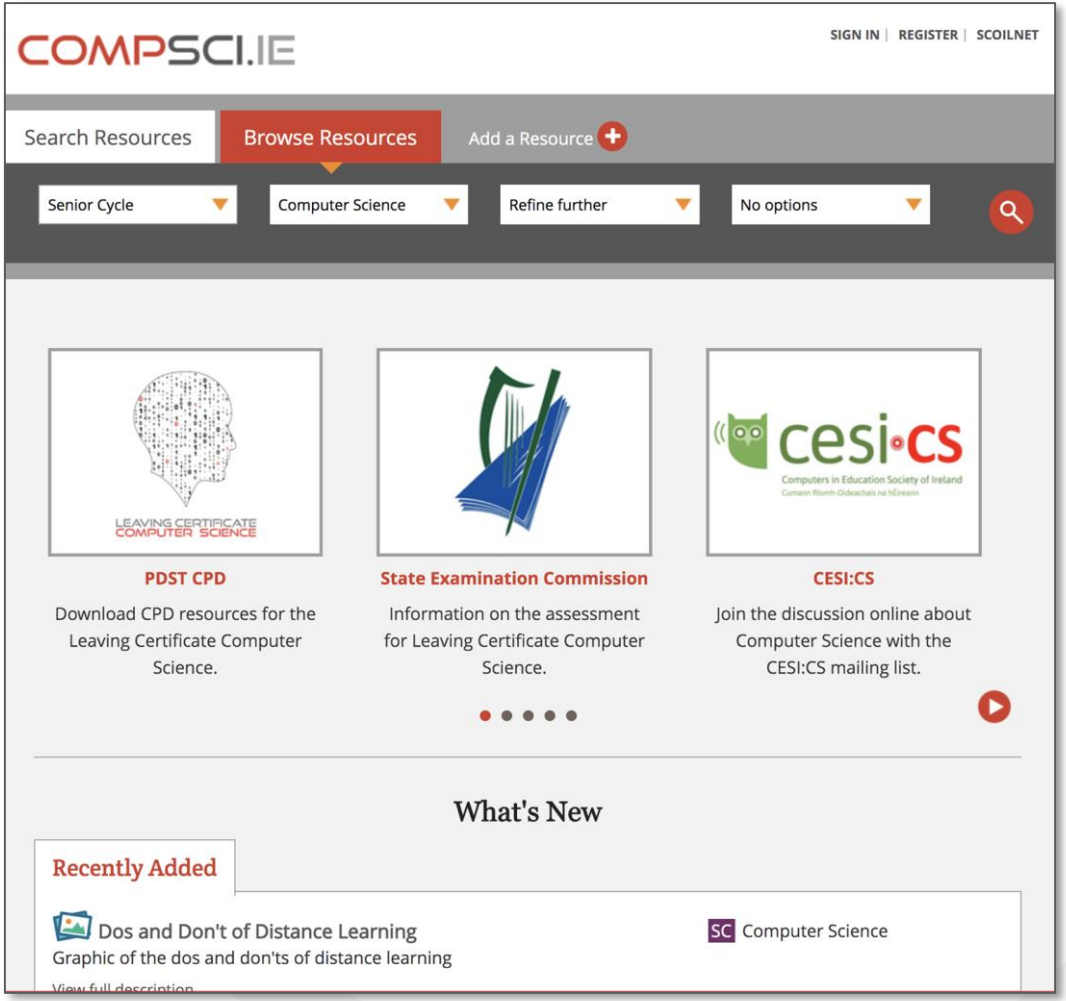






Plus lots more ….

COMPSCI.IE


COMPSCI.IE
SIGN IN | REGISTER | SCOILNET
Search Resources
Browse Resources
Add a Resource
Senior Cycle
Computer Science
Refine further
No options
LEAVING CERTIFICATE COMPUTER SCIENCE
PDST CPD
Download CPD resources for the Leaving Certificate Computer Science.
State Examination Commission
Information on the assessment for Leaving Certificate Computer Science.
cesi·cs
Computers in Education Society of Ireland
CESI:CS
Join the discussion online about Computer Science with the CESI:CS mailing list.
What's New
Recently Added
Dos and Don't of Distance Learning
Graphic of the dos and don'ts of distance learning
View full description
SC Computer Science

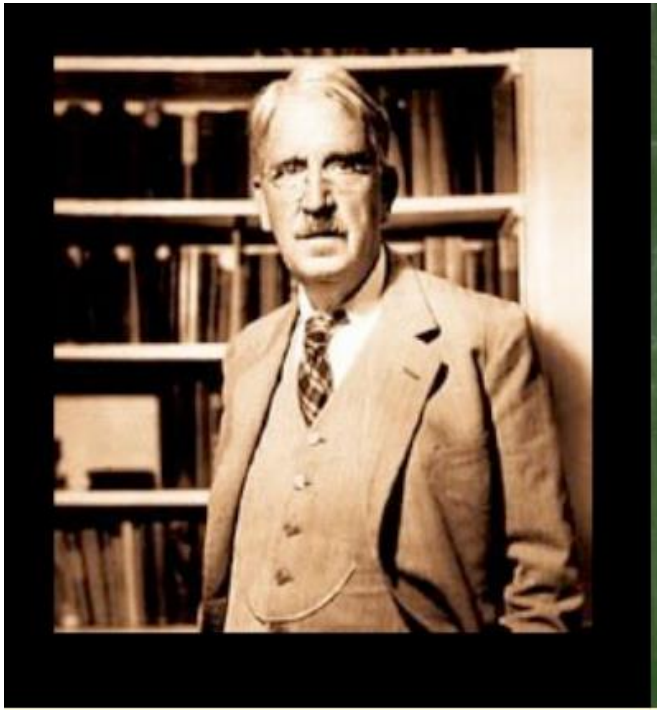
slack

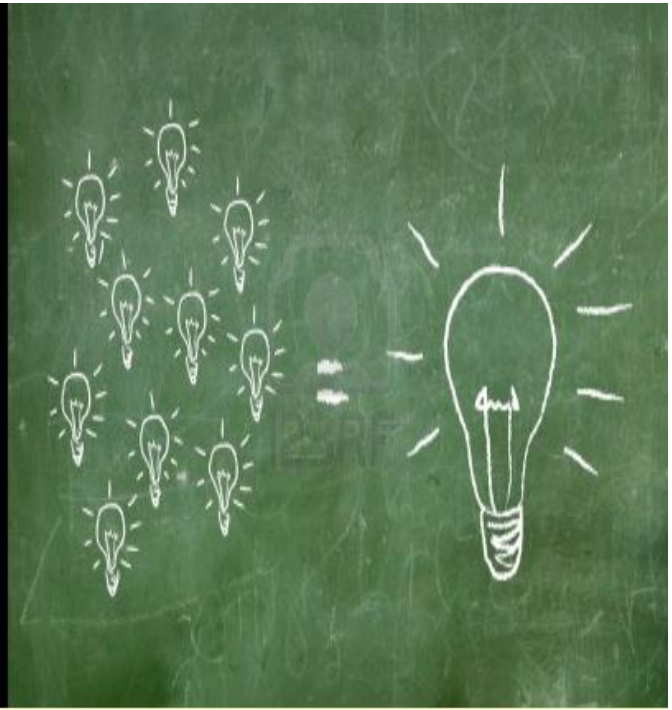
Workspace: pdst_phase_3

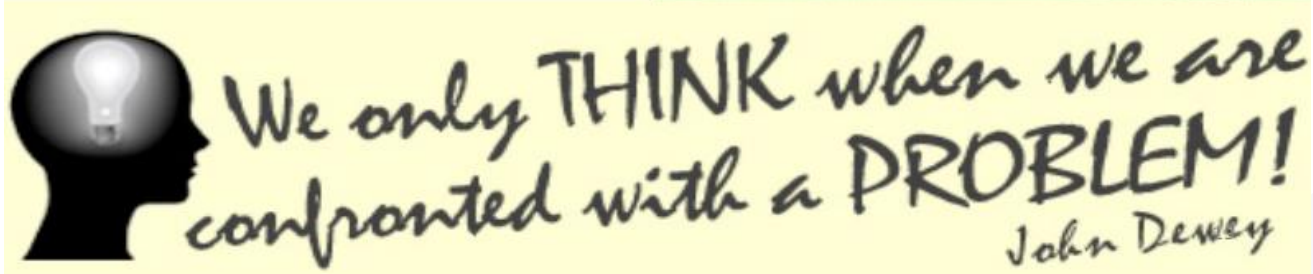
computerscience@pdst.ie

?

@PDSTcs






We only THINK when we are confronted with a PROBLEM!
John Dewey

An Roinn Oideachais
Department of Education