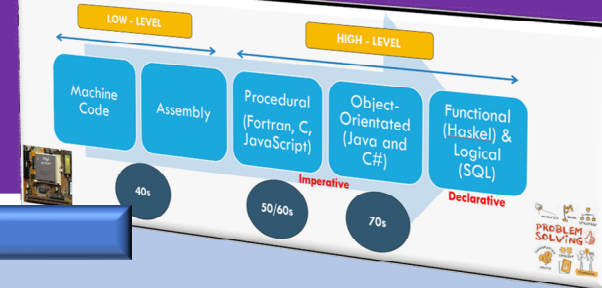


The Importance of Making Programming Paradigms Pop!



1 Rationale

Most programming languages have more in common than they do differences, and can be classified into relatively few programming paradigms, or language styles.

This study highlights the importance of 'programming paradigms' as an underpinning concept that should be emphasised and given substantial time in support of Tim Oate's (2011) vision of 'mastery', which inspired the 2010 National Curriculum (NC) Review and the DfE's (2014) A-Level Computer Science (CS) Specification.

2 Theory

ADVANCE ORGANIZERS IN LEARNING AND RETENTION (Ausubel, 1960)

"The National Curriculum should be concept-led and knowledge-led, not context led." (Oates, 2011, p. 132)

Vygotsky's (1978) ZPD argues that what a child can do with support today, will be mastered tomorrow.

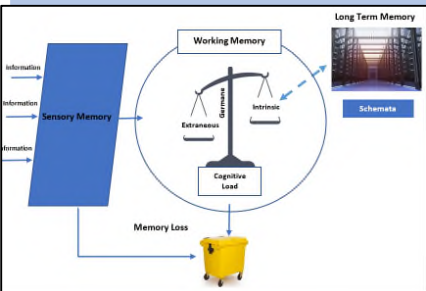


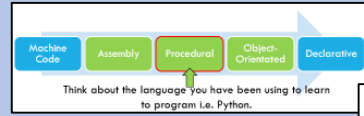
Fig 1. Waldron-Healy's (2021) illustration of Sweller et al's (2019) update to the categorisation of germane load.

Fig. 2 William and Thompson's (2008) enactment of Formative Assessment strategies by different agents of learning.

	Where the learner is going	Where the learner is right now	How to get there
Teacher	1 Clarifying learning intentions and criteria for success	2 Engineering effective classroom discussions and other learning tasks that elicit evidence of student understanding	3 Providing feedback that moves learners forward
Peer	Understanding and sharing learning intentions and criteria for success	4 Activating students as instructional resources for one another	
Learner	Understanding learning intentions and criteria for success	5 Activating students as the owners of their own learning	

3 Practice

TECHNIQUE 16 BEGIN WITH THE END (Doug Lemov, 2015)



DfE: ASSIGN A TERM TO ONE OR BOTH

LOW LEVEL PROGRAMMING LANGUAGES	HIGH LEVEL PROGRAMMING LANGUAGES
Iteration	Translator
Mnemonic	Python
Imperative	1:1
High Abstraction	Maximises
Executable File	Compiler
CPU Agnostic	Sequence
	Assembler
	Declarative
	SQL
	Machine Code
	Interpreter
	Selection

LESSON OBJECTIVES

- Understand the need for and features of a variety of programming paradigms.
- Describe key features of procedural programming languages.
- Make connections between different procedural programming languages.

ACCURATE AND PRODUCTIVE USE OF ASSESSMENT (TSG)

Support:
(a) What sets it apart from assembly or OO?
(b) Reflect on your studies of 'Computational Thinking'.

4 Critical Discussion

- Oates (2014) warns that a lack of coherence between curriculum content, assessment and accountability arrangements, governance and professional development will inhibit the desired shift from 'surface' to 'deep learning' (Black and Wiliam, 1998).
- A shortage of specialist and/or suitably skilled computer science teachers (Kemp and Berry, 2019; NCCE, 2020) is likely to encourage a focus on context i.e. specific programming languages, to the detriment of underpinning concepts and knowledge.

History Repeating Itself?
Today the picture is very different: with the best of intentions, we have lost the "how it works" part, in favour of "how to use it". (CAS, 2012, p. 2)

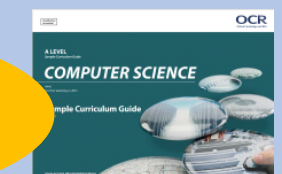


Fig 3.0 OCR A-Level CS Curriculum Planning Guide (2016)

5 Conclusion

The time I devoted to delivering programming paradigms as an underpinning concept facilitated my use of formative assessment, and in turn raised standards of achievement and engagement.

Moving forwards, I would aim to:

- refine my use of PRIMM, as pioneered by Sentance and Waite (2017), to lessen cognitive load associated with the intrinsic difficulty and extraneous presentation of learning activities designed to foster mastery of programming paradigms (Sweller, Van Merriënboer and Paas, 1998).
- trial paired-programming to facilitate peer- and self- assessment as advocated by Wiliam and Thompson (2008) to promote the learning of all students, and encourage female students to persist in their studies of CS (Werner, 2004).

6 References

AQA (2015) AS and A-Level Computer Science Specification: 7516/7517.
 Ausubel, D. (1960) The use of advance organizers in the learning and retention of meaningful verbal material, *Journal of Educational Psychology*.
 CAS (2012) Computer Science: A curriculum for schools.
 DfE (2013) National curriculum in England: computing programmes of study - GOV.UK.
 DfE (2014) GCE AS and A level subject content for computer science.
 Doug Lemov (2015) *Teach Like a Champion*. 2.0. San Francisco: Jossey-Bass.
 Gove, M. (2012) 'ICT in the National Curriculum', in 2012 BETT Show. DfE.
 De Groot, A. (1965) Thought and Choice in Chess.
 Kaczmarczyk, L. et al. (2010) Identifying Student Misconceptions of Programming.
 Kemp, P. E. and Berry, M. G. (2019) The Roehampton Annual Computing Education Report Pre-release snapshot from 2018. London.
 Livingstone, I. and Hope, A. (2011) Next Gen: Transforming the UK into the world's leading talent hub for the video games and visual effects industries.
 Miller, G. (1994) 'The magical number seven, plus or minus two: Some limits on our capacity for processing information.', *Psychological Review*, 101(2), pp. 343-352.
 NCCE (2020) Impact Report.
 Oates, T. (2011) 'Could do better: using international comparisons to refine the National Curriculum in England', *Curriculum Journal*, 22(2), pp. 121-150.
 OCR (2016) Sample Curriculum Planning Guide: H446.
 OCR (2018) GCE Computer Science: OCR Report to Centres.
 OCR (2020) Qualification Accredited A LEVEL Clarification Guide: H446.
 Ofsted (2011) ICT in schools 2008-2011.
 Ofsted (2019) School inspection handbook.
 Sentance, S. and Waite, J. (2017) 'PRIMM: Exploring pedagogical approaches for teaching text-based programming in school', in *ACM International Conference Proceeding Series*, pp. 113-114.
 Sweller, J. (1988) 'Cognitive load during problem solving: Effects on learning', *Cognitive Science*, 12(2), pp. 257-285.
 Sweller, J., Van Merriënboer, J. J. G. and Paas, F. G. W. C. (1998) 'Cognitive Architecture and Instructional Design', *Educational Psychology Review*, 10(3), pp. 251-296.
 The Royal Society (2012) Shut down or restart? The way forward for computing in UK schools.
 Vygotsky, L. (1978) 'Mind in Society: The Development of Higher Psychological Processes', in.
 Waldron-Healy, G. (2021) 'Implications of current theories of cognitive development in teaching and learning'.
 Werner, L. L. (2004) Pair-Programming Helps Female Computer Science Students, *ACM Journal of Educational Resources in Computing*.
 Wiliam, D. and Thompson, M. (2008) 'What is formative assessment?', in *Future of assessment: Shaping teaching and learning*, pp. 10-26.
 Wing, J. M. (2006) 'Computational Thinking', *Communications of the ACM*, 49(3), pp. 33-35.