DIGITAL TECHNOLOGIES: KNOWLEDGE OF COMPUTER SCIENCE AND SOFTWARE ENGINEERING

Computer science and software engineering refers to a group of concepts associated with the discipline of computer science and how they are applied in user interfaces.

Initially students learn about basic concepts of algorithms, programming language and user interface. Students progress to learning about tractability, data representations, coding, usability heuristics, formal specification of the syntax of programming languages, and software development methods.

	LEVEL 6	LEVEL 7	LEVEL 8
LO	*Demonstrate understanding of basic concepts in computer science and software engineering*	*Demonstrate understanding of advanced concepts in computer science and software engineering*	*Demonstrate understanding of areas of computer science*
TEACHER GUIDANCE	To support students to develop understandings about the basic concepts in computer science and software engineering at level 6, teachers could: • Ensure students understand the concept of an algorithm vs. a program, and that there are different costs for different algorithms for the same task. This could be illustrated with searching (linear and binary) and/or sorting. • Provide students with an opportunity to understand the programming language concepts of: high level languages, machine languages, interpretation and compilation; and the idea that programming languages are precise. • Guide students to informally critique user interfaces based on personal experience rather than using heuristics – for example, identify a frustrating user interface and explain why it was difficult to use.	To support students to develop understandings about the advanced concepts in computer science and software engineering at level 7, teachers could: • Ensure students understand the concepts of complexity and tractability– the idea that some problems are inherently difficult to solve on a computer. • Provide students with an opportunity to understand how various kinds of data can be represented using bits. • Provide students with an opportunity to understand how coding for compression, error control or encryption enable technologies, for example, mp3 players, reliable storage and communication, e-commerce. • Guide students to evaluate a Human-Computer interface in terms of simple usability heuristics – Nielsen's usability heuristics would be a suitable framework to use.	To support students to develop understandings about the areas of computer science, at level 8, teachers could: • Ensure students understand key problems in selected areas of computer science. Selected areas include: formal languages, network communication protocols, complexity and tractability, Intelligent systems, software engineering, graphics and visual computing. • Ensure students have a framework for investigating areas of computer science that includes understanding the key problems in that area, practical applications, and the use of algorithms and/or techniques from that area. • Provide students with an opportunity to understand how key algorithms and techniques address key problems in selected areas of computer science. • Provide students with an opportunity to understand examples of practical applications in selected areas. • Guide students to evaluate the effectiveness of algorithms, techniques, or applications from selected areas. • Guide students to understand solved and unsolved problems in selected area of computer science. • Support students to practice report writing, including ways to structure a report, and literacy strategies to support report writing in a way that will allow students to explain, discuss, and evaluate.
INDICATORS	Students can: • explain how algorithms are distinct from related concepts such as programs and informal instructions • compare and contrast the concepts of algorithms, programs, and informal instructions • determine and compare the costs of two different iterative algorithms for the same problem of size n • compare and contrast high level and low level (or machine) languages, and explaining different ways in which programs in a high level programming language are translated into a machine language • discuss how different factors of a user interface contribute to its usability by comparing and contrasting related interfaces.	Students can: • compare and contrast different ways in which different types of data can be represented using bits and discuss the implications • discuss how a widely used technology is enabled by one or more of compression coding, error control coding, and encryption enable • suggest improvements to a given human-computer interface based on an evaluation in terms of simple usability heuristics.	Students can: • discuss solved and unsolved problems in the selected areas of computer science • discuss examples to explain how practical applications from selected areas of computer science use algorithms and/or techniques from these areas • explain how key algorithms or techniques address key problems in selected areas of computer science • evaluate the effectiveness of algorithms and/or techniques applied in selected areas of computer science.
AS	**AS91074 Digital Technologies 1.44** *Demonstrate understanding of basic concepts from computer science*	**AS91371 Digital Technologies 2.44** *Demonstrate understanding of advanced concepts from computer science*	**AS91636 Digital Technologies 3.44** *Demonstrate understanding of areas of computer science*
	Level 1 Digital Technologies standards & assessment resources	Level 2 Digital Technologies standards & assessment resources	Level 3 Technology achievement standards & assessment resources DRAFT