DIGITAL TECHNOLOGIES: KNOWLEDGE OF DIGITAL INFORMATION MANAGEMENT

Knowledge of digital information management focuses on how information is managed at both an individual user level and with shared information within an organisation. Initially students learn about basic concepts of information management in relation to producing digital information outcomes. This includes understanding the key features of operating systems and common application software, file management procedures, and ethical issues related to the management of information. Students progress to learning about complex concepts of information systems within organisations. This includes explaining the interaction between the main components of an information system used in an organisation, discussing the nature and value of information to an organisation, discussing the characteristics of good information, and end-user considerations, and discussing the implications of security management for information systems.

	LEVEL 6	LEVEL 7	LEVEL 8
LO	*Demonstrate understanding of basic digital information management tools and systems*	*Demonstrate understanding of advanced digital information management tools and systems*	*Demonstrate understanding of complex concepts of information systems in an organisation*
TEACHER GUIDANCE	To support students to develop understandings about basic digital information management tools and systems at level 6, teachers could: • Provide students with opportunities to explain how application software and operating system software interact to manage information. • Provide students with a way of selecting and justifying the selection of application software to perform a task to manage information. • Provide students with opportunities to explain the purpose and conventions of file management procedures and the use of storage devices. • Provide students with opportunities to compare and contrast the use of different file types for different purposes • Provide students with opportunities to describe ethical issues related to management of information. • Guide students on how to prepare reports including ways to structure a report and literacy strategies to support report writing in a way that will allow students to identify, describe, explain, justify, and compare and contrast. • Provide opportunities for students to practice report writing including acknowledging sources and bibliographies.	To support students to develop understandings about advanced digital information management tools and systems at level 7, teachers could: • Guide students on how to research the information management issues related to shared information within an organisation. • Provide students with opportunities to explain file management considerations related to shared information and the related procedures and conventions for privacy and permission. • Provide students with opportunities to discuss ethical and legal issues related to shared information within an organisation. • Provide students with opportunities to evaluate backup procedures and conventions for information systems within an organisation. • Provide students with opportunities to evaluate the effectiveness of information systems for managing shared information within an organisation. • Ensure students understand the requirement at this level to look at information management issues related to shared information and information systems within an organisation • Ensure students have access to a suitable organisation to use for the case study. • Support students to prepare reports including ways to structure a report and literacy strategies to support report writing in a way that will allow students to describe, explain, discuss, and evaluate. • Ensure students have opportunities to practice report writing including acknowledging sources and bibliographies.	To support students to develop understandings about complex concepts of information systems in an organisation, at level 8, teachers could: • Guide students on how to research the information systems within an organisation. • Provide students with opportunities to explain the interaction between the main components (hardware, software, data, procedures, and people) of an information system used in an organisation. • Provide students with opportunities to discuss the nature and value of information to an organisation including being able to discuss the differences between data, information, and knowledge. • Provide students with opportunities to discuss the characteristics of 'good' information (such as accuracy, timeliness, relevance, appropriate quantity, economical etc) and evaluate the trade-offs between the characteristics of good information in an organisation. • Provide students with opportunities to discuss the impact on, and influence of, end-user considerations (such as user consultation, ease-of-use, user interface design, work procedures, implementation issues, training) on information systems in an organisation. • Provide students the opportunity to discuss the implications of security management for information systems (including evaluating trade-offs between security management and end-user considerations within an organisation). • Ensure students understand the requirement at this level to look at an information systems within an organisation, and that when looking at an information system they need to look at hardware, software, data, procedures, and people. • Ensure students have access to a suitable organisation to use for the case study. • Support students to prepare reports 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. • Ensure students have opportunities to practice report writing including acknowledging sources and bibliographies.
INDICATORS	Students can: • identify and describe key features of operating systems and common application software as they relate to the management of information • identify and describe file management procedures, and explain the purpose and conventions of file management procedures and use of storage devices • describe ethical issues related to management of information (eg, copyright, privacy, file security, appropriateness of the material in its context) • explain the purpose of operating systems and the purposes of common application software to manage information and how application software and operating system software interact to manage information • justify the selection of application software to perform a task to manage information • compare and contrast the use of different file types for different purposes (eg, pdf versus doc, jpeg versus bmp).	Students can: • explain the file management considerations related to shared information • explain the role of an information system for managing shared information within an organisation, and discuss the advantages and disadvantages of an information system for managing shared information within an organisation • identify the input, storage, retrieval and manipulation of data within an information system • discuss ethical and legal issues related to shared information within an organisation and the implications for procedures and conventions for privacy and permission • explain the implications of back up procedures and conventions for information systems within an organisation, and evaluate the backup procedures and conventions for information systems used within an organisation • evaluate procedures and conventions for privacy and permissions used within an organisation • evaluate the effectiveness of an information system for managing shared information within an organisation.	Students can: • explain the interaction between the main components of an information system used in an organisation • explain the nature of information and discuss: differences between data, information and knowledge; the nature and value of information in an organisation; and how information systems add value to an organisation • discuss the characteristics of 'good' information, and evaluate the trade-offs between the characteristics of good information (eg, timeliness vs. accuracy) in an organisation • explain the importance of end-user considerations in information systems, and discuss the impact on and influence of end-user considerations on information systems in an organisation • explain security management for information systems and discuss the implications of security management for information systems • evaluate the trade-offs between security management in an information system and end-user considerations in an organisation.
AS	**AS91070 Digital Technologies 1.40** *Demonstrate understanding of basic concepts of information management*	**AS91367 Digital Technologies 2.40** *Demonstrate understanding of advanced concepts relating to managing shared information within information systems*	**AS91632 Digital Technologies 3.40** *Demonstrate understanding of complex concepts of information systems in an organisation*
	Level 1 Digital Technologies standards & assessment resources	Level 2 Digital Technologies standards & assessment resources	Level 3 Technology achievement standards & assessment resources DRAFT

DIGITAL TECHNOLOGIES: CREATE A DIGITAL INFORMATION OUTCOME

Apply digital information management tools to create a digital information outcome requires students to create a digital information outcome that involves manipulating and combining data from more than one application. The specifications for the digital information outcome, software and techniques to be used need to be determined prior to the outcome being made.

When creating digital information outcomes students will use appropriate techniques and data integrity and testing procedures. Students will apply appropriate file management procedures, design elements, and formatting techniques. Students will consider their legal, ethical, and moral responsibilities when developing digital information outcomes.

Initially students learn to perform a set of basic procedures, as instructed, to produce a digital information outcome that involves manipulating and combining data from at least two applications out of word processing, spreadsheets, database, and presentation software. Students should progress to using complex procedures to design and produce a database application with dynamically linked data.

	LEVEL 6	LEVEL 7	LEVEL 8
LO	*Implement basic procedures to create a digital information outcome*	*Implement advanced procedures to create a digital information outcome with dynamically linked data*	*Implement complex procedures to develop a relational database embedded in a specified digital outcome*
TEACHER GUIDANCE	To support students to implement basic procedures to produce a digital information outcome, at level 6, teachers could: • Provide an opportunity for students to learn and practice a range of basic procedures in at least two software application types from word processing, spreadsheets, database, and presentation. • Provide a brief for a specified digital information outcome, or guide students to develop their own brief. • Ensure students are aware of the requirement at this level to produce an outcome that results from manipulating and combining data from at least two of the software application types listed above. • Support students to apply appropriate file management procedures when developing digital information outcomes • Support students to apply design elements and formatting techniques as they develop digital information outcomes • Support students to apply data integrity and testing procedures as they develop digital information outcomes. • Support students to understand their legal, ethical, and moral responsibilities when developing digital information outcomes.	To support students to implement advanced procedures to produce a digital information outcome, at level 7, teachers could: • Provide an opportunity for students to learn and practice a range of advanced procedures in a database and at least one other software application. • Provide a brief for a specified digital information outcome, or guide students to develop their own brief. • Ensure students are aware of the requirement at this level to produce an outcome that integrates data from a database and at least one other application using dynamic linking. • Ensure students apply design elements and formatting techniques accurately and independently as they develop digital information outcomes. • Ensure students apply data integrity and testing procedures as they develop digital information outcomes. • Ensure students understand their legal, ethical, and moral responsibilities when developing digital information outcomes in terms of the social implications of the outcome within an organisation and the wider community.	To support students to implement complex procedures to develop a relational database embedded in a specified digital outcome, at level 8, teachers could: • Provide an opportunity for students to learn and practice a range of complex procedures in a relational database. • Provide opportunities for students to develop skills in Entity Relationships (ERD) and normalisation. • Ensure students know how to allow data in at least one database table to be changed from another application. • Provide a brief for a specified digital outcome, or guide students to develop their own brief. • Ensure students are aware of the requirement at this level to produce a relational database with a well organised table structure and where data in at least one database table is changed using another application. • Ensure students apply design elements and formatting techniques as appropriate to the outcome they develop to create an interface for navigation, collection and display of data which demonstrates usability considerations. • Ensure students apply data integrity and testing procedures as they develop the digital outcomes. • Ensure students can apply data access permissions which show they understand their legal, ethical, and moral responsibilities when developing digital information outcomes.
INDICATORS	Students can: • select appropriate software and apply basic techniques to produce a specified digital information outcome that meets specifications and involves manipulating and combining data from at least two software applications out of word processing, spreadsheets, database, and presentation • apply appropriate file management procedures • apply appropriate design elements and/or formatting techniques • show accuracy in the application of techniques and testing procedures • show independence with regard to decision making in the application of techniques, and testing procedures. • undertake techniques and testing procedures in a manner that economises the use of resources in the outcome's production and its use • follow legal, ethical and moral responsibilities as appropriate to the outcome.	Students can: • apply advanced techniques to produce a specified digital information outcome that meets specifications and integrates data from a database and one other application using dynamic linking • apply appropriate design elements and/or formatting techniques • show accuracy in the application of techniques, design elements, and testing procedures • show independence with regard to decision making in the application of techniques, design elements, and testing procedures. • undertake techniques and testing procedures in a manner that economises the use of resources in the outcome's production and its use • follow legal, ethical and moral responsibilities as appropriate to the outcome by considering the social implications of the outcome within an organisation and the wider community.	Students can: • apply complex techniques to design and produce a relational database embedded in a specified digital outcome that has a well-organised table structure use features of the software to allow data in at least one database table to be changed using another application • apply design elements and formatting techniques as appropriate to the outcome to create an interface for navigation, collection and display of data which demonstrates usability considerations • show accuracy in the application of techniques and testing procedures • show independence with regard to decision making in the application of techniques, and testing procedures. • undertake techniques and testing procedures in a manner that economises the use of resources in the outcome's production and its use • apply data access permissions which follow legal, ethical and moral responsibilities as appropriate to the outcome.
AS	**AS91071 Digital Technologies 1.41** *Implement basic procedures to produce a specified digital information outcome*	**As91368 Digital Technologies 2.41** *Implement advanced procedures to produce a specified digital information outcome with dynamically linked data*	**AS91633 Digital Technologies 3.41** *Implement complex procedures to develop a relational database embedded in a specified digital outcome*
	Level 1 Digital Technologies standards & assessment resources	Level 2 Digital Technologies standards & assessment resources	Level 3 Technology achievement standards & assessment resources DRAFT

DIGITAL TECHNOLOGIES: KNOWLEDGE OF DIGITAL MEDIA

Knowledge of digital media focuses on understanding of concepts of digital media that need to be considered when developing digital media outcomes. Initially students learn about basic concepts of digital media. These basic concepts include such things as the media types, software resources, and techniques used to create digital media outcomes. They also include such things as influence of design elements, communication purpose, and ethics are considered when developing digital media outcomes. Students progress to learning about complex concepts of digital media such as those tools and techniques used to present content across multiple outcomes, application of digital media standards and conventions, asset management, file management, naming conventions, and legal, ethical, and moral considerations in relation to the requirements of digital media outcome within the wider community.

	LEVEL 6	LEVEL 7	LEVEL 8
LO	*Demonstrate understanding of basic concepts in digital media*	*Demonstrate understanding of advanced concepts in digital media*	*Demonstrate understanding of complex concepts of digital media*
TEACHER GUIDANCE	To support students to develop understandings about basic concepts in digital media at level 6, teachers could: • Provide students with the opportunity to explore a range of digital media outcomes. • Ensure the digital media outcomes students investigate demonstrate an integration of media types and are of sufficient rigour for this level (see indicators below). • Provide opportunity for students to explain how basic tools and techniques have been used to create digital media outcomes. • Guide students to consider why ethical considerations are important when developing digital media outcomes • Guide students on how to prepare reports including ways to structure a report . • Support students to develop literacy strategies that assist them to write reports in a way that will allow them to identify, describe, explain, and discuss.	To support students to develop understandings about the advanced concepts in digital media at level 7, teachers could: • Provide students with the opportunity to explore a range of digital media outcomes. • Ensure the digital media outcomes students investigate demonstrate an integration of media types and are of sufficient rigour for this level (see indicators below). • Provide opportunity for students to explain how advanced tools and techniques have been used to create digital media outcomes. • Provide opportunity for students to discuss the implications of adhering to digital media standards and conventions whilst developing digital media outcomes. • Provide opportunity for students to discuss the importance of asset management and file management whilst developing digital media outcomes. • Guide students to consider the legal, ethical, and moral responsibilities to the wider community when developing digital media outcomes. • Guide students to understand the importance of appropriate data integrity and testing procedures whilst developing digital media outcomes. • Support students to develop literacy strategies that assist them to write reports in a way that will allow them to identify, describe, explain, and discuss	To support students to develop understandings about the complex concepts of digital media at level 8, teachers could: • Provide students with the opportunity to explore a range of digital media outcomes. • Ensure the digital media outcomes students investigate demonstrate an integration of media types and are of sufficient rigour for this level (see indicators below). • Provide opportunity for students to discuss design elements associated with complex digital media outcomes. • Ensure students understand why content needs to be manipulated to be used across media types. • Provide opportunity for students to explain how complex tools and techniques have been used to create complex digital media outcomes. • Provide opportunity for students to discuss the needs as well as the implications of adhering to digital media standards and conventions to present content in complex digital multiple outcomes. • Guide students to explain the legal, ethical, and moral responsibilities to the wider community when developing digital media outcomes. • Guide students to understand the importance of appropriate data integrity and testing procedures whilst developing digital media outcomes. • Support students to develop literacy strategies that assist them to write reports in a way that will allow them to, explain, and discuss the various concepts of digital media.
INDICATORS	Students can: • identify the digital media types in a digital media outcome • describe the software resources used to create a digital media outcome • describe basic techniques used to create a digital media outcome • describe design elements in a digital media outcome • describe the communication purpose of a digital media outcome, and discuss why distinguishing characteristics of a digital media outcome support its communication purpose • explain how software resources and techniques used affected the technical quality of a digital media outcome • explain why ethical considerations were important in the creation of a digital media outcome • discuss why software resources and techniques were used to create a digital media outcome and how they affected its technical quality.	Students can: • discuss why advanced tools and techniques have been used to create, edit and integrate digital media outcomes and how their use have enhanced the outcome • explain the digital media standards and conventions used to produce digital media outcomes, and discuss the positive and negative implications of adhering to these standards and conventions when developing digital media outcomes • explain how asset management and file management are applied in the development of digital media outcomes, and discuss the importance of effective and appropriate asset management and file management in the development of digital media outcomes • explain legal, ethical and moral considerations in relation to the requirements of digital media outcomes in the wider community • explain the data integrity and testing procedures used to ensure a digital media outcome meets the specifications, and how they were applied • discuss the importance of appropriate data integrity and testing procedures in the development of digital media outcomes • evaluate how the application of advanced tools, techniques, standards and conventions affect the quality of digital media outcomes • discuss the relationship between standards and conventions, and legal, ethical and moral considerations in relation to the requirements of digital media outcomes.	Students can: • explain the complex tools and techniques used to present content across multiple digital media outcomes • explain the importance of selecting appropriate digital media software applications and specific features to present content across multiple outcomes • discuss design elements associated with complex digital media outcomes • explain the implications of adhering to digital media standards and conventions to present content across multiple digital media outcomes, and discuss the needs as well as the positive and negative implications of adhering to these standards and conventions when developing digital media outcomes • explain the data integrity and testing procedures used to ensure a digital media outcome meets specifications • explain the legal, ethical and moral considerations in relation to the requirements of a digital media outcome in the wider community • discuss the relationship between the tools, techniques, design elements, legal, ethical and moral considerations in relation to the requirements of the outcome
AS	**AS91072 Digital Technologies 1.42** *Demonstrate understanding of basic concepts of digital media*	**AS91369 Digital Technologies 2.42** *Demonstrate understanding of advanced concepts of digital media*	**AS91634 Digital Technologies 3.42** *Demonstrate understanding of complex concepts of digital media*
	Level 1 Digital Technologies standards & assessment	Level 2 Digital Technologies standards & assessment resources	Level 3 Technology achievement standards & assessment resources DRAFT

DIGITAL TECHNOLOGIES: CREATE A DIGITAL MEDIA OUTCOME

Create a digital media outcome requires students to construct a digital media outcome that integrates media types and incorporates original content. The specifications for the digital media outcome, software and techniques to be used need to be determined prior to the outcome being made.

Initially students learn to perform a set of techniques, as instructed, to produce a digital media outcome. Students should progress to integrating digital media types using complex tools and techniques (eg, *Web design*: HTML/CSS, scripting dynamic data handling, interaction between user and content, multiple device outputs; *Desktop publishing*: interactivity, form elements, chapters and sections: *Motion graphics*: Complex transitions, multiple tracks, post processing; *Audio*: multiple tracks, manipulating multiple tracks, overlays, equalising; *Image manipulation:* colour histograms and adjustments, non destructive editing, pen tools and paths, filter effects, graphic optimisations, colour management and printing, automation/scripts) when constructing original digital media outcomes such as an *animation or multi-page website* that integrates student created graphics, video and/or audio; an edited, student-produced *movie* integrating student created soundtrack, graphics and/or animation; a *multi-page desktop published document* integrating student created graphics and/or still images.

	LEVEL 6	LEVEL 7	LEVEL 8
LO	*Implement basic procedures to create a digital media outcome*	*Implement advanced procedures to create a digital media outcome*	*Implement complex procedures to create a digital media outcome*
TEACHER GUIDANCE	To support students to implement basic procedures to create a digital media outcome, at level 6, teachers could: • Provide opportunity for students to explore and develop technical expertise with digital media tools. • Provide opportunity for students to apply an understanding of digital media to design and create a number of different digital media outcomes using a variety of digital media technologies. • Provide opportunity for students to develop an understanding about the legal, ethical and moral responsibilities as appropriate to a digital media outcome. • Provide opportunity for students to develop an understanding about, and apply data integrity and testing procedures that ensure a digital media outcome meets brief specifications • Provide opportunity for students to interpret the needs of a situation to publish a successful and appropriate digital media outcome.	To support students to implement advanced procedures to create a digital media outcome, at level 7, teachers could: • Provide opportunity for students to apply advanced tools and techniques to enhance the presentation of digital media content. • Guide students to select digital media software applications and specific features when creating, editing and integrating digital media. • Provide opportunity for students to identify and apply file management techniques to successfully publish digital media content. • Provide opportunity for students to input, manipulate and test digital media data, and ensure its compliance with current web standards or other digital media compliance requirements. • Provide opportunity for students to apply processes for media input, editing, testing, and publishing techniques that ensure data integrity; and consider legal, ethical and moral responsibilities that ensure a digital media outcome addresses a brief's specifications. • Provide opportunity for students to create virtual, incomplete or new realised digital media outcomes in a manner that economises the use of resources in production and ensure a digital media outcome's usability, for example, timely fashion, optimisation of tool selection and use.	To support students to implement complex procedures to create a digital media outcome, at level 8, teachers could: • Provide opportunity for students to select digital media software applications and specific features to create, edit and integrate digital media types. • Provide opportunity for students to apply complex tools and techniques to create adigital media outcome. • Guide students to apply processes for media input, editing, testing, and publishing techniques that ensure data integrity and consider legal, ethical and moral responsibilities that ensure a digital media outcome addresses a brief's specifications. • Provide the opportunity for students to create digital media outcomes in a manner that economises the use of resources in production and ensures a digital media outcome's usability.
INDICATORS	Students can: • use appropriate features of digital media software to edit and integrate digital media types to create a digital media outcome • apply formatting techniques, design elements, and data integrity and testing procedures, to ensure the outcome meets the specifications • follow legal, ethical, and moral responsibilities as appropriate to the outcome • show accuracy and independence in the application of techniques and testing procedures • undertake techniques and testing procedures in a manner that economises the use of resources in a digital media outcome's production and use.	Students can: • select software based on the features of the program(s) that enable media types to be created, edited and integrated • use advanced tools and techniques to edit and integrate digital media types to create a digital media outcome • apply advanced formatting techniques, design elements, and data integrity and testing procedures, to ensure a digital media outcome meets the specifications • follow legal, ethical, and moral responsibilities as appropriate to a digital media outcome • show accuracy and independence in the application of advanced tools, techniques and testing procedures • apply tools and techniques and testing procedures in a manner that economises the use of resources in a digital media outcome's production and usability.	Students can: • select appropriate digital media software applications and specific features to create, edit and integrate digital media types • accurately and independently apply a set of complex tools and techniques, as appropriate to the media, to create a digital media outcome • accurately and independently apply data integrity and testing procedures • follow legal, ethical and moral responsibilities as appropriate to the outcome • effectively undertake techniques and procedures in a manner that economises the use of resources in a digital media outcome's production and usability.
AS	**AS91073 Digital Technologies 1.43** *Implement basic procedures to produce a specified digital media outcome*	**AS91370 Digital Technologies 2.43** *Implement advanced procedures to produce a specified digital media outcome*	**AS91635 Digital Technologies 3.43** *Implement complex procedures to produce a specified digital media outcome*
	Level 1 Digital Technologies standards & assessment resources	Level 2 Digital Technologies standards & assessment resources	Level 3 Technology achievement standards & assessment resources DRAFT

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

DIGITAL TECHNOLOGIES: DESIGN A SOFTWARE PROGRAM STRUCTURE

Design a software program focuses on designing the structure of a software programs. Initially students learn to specify variables and their data types, construct flexible and robust plans, and determine structures that combine well-chosen actions, conditions and control structures that provide well-structured logical solution to tasks. They establish sets of test cases with expected, boundary and invalid input for testing programs. Students progress to designing the structure of a complex software program where the plan has a modular structure, an indexed data structure, input and output, and procedural structures that combine sequential, conditional, and iterative structures. By level 8 students should be using an Integrated Development Environment (IDE) to develop code following a disciplined development process with cycles of incremental development and testing.

	LEVEL 6	LEVEL 7	LEVEL 8
LO	*Demonstrate ability to design the structure of a basic software program*	*Demonstrate ability to design the structure of an advanced software program*	
TEACHER GUIDANCE	To support students to develop an ability to design the structure of a basic software program at level 6, teachers could: • Guide students on how to specify variables and their data types. • Guide students to independently construct flexible and robust plans for basic programs that include using actions, conditions and control structures such as: checking input data for validity; correctly handling expected, boundary and invalid inputs; and using constants, variables and derived values in place of literals. • Guide students to set out program codes clearly and to document programs with comments. • Guide students on how to specify procedural structures that combine well-chosen actions, conditions and control structures that constitute well-structured logical solution to tasks which have no unnecessary duplication or repetition. • Guide students specify comprehensive sets of test cases with expected, boundary and invalid input for testing programs.	To support students to develop an ability to design the structure of an advanced software program at level 7, teachers could: • Guide students on how to specify well-chosen scopes for variables, their scopes and data types • Guide students on how to specify indexed data structures • Guide students on how to specify modular structures for programs with well-chosen parameters, including details of procedural structures of modules, that constitute well-structured logical decomposition of tasks • Guide students on how to specify a comprehensive set of expected, boundary and exceptional input cases for testing programs. • Guide students on how to specify variables, constants, and derived values effectively so as to maximise the flexibility and robustness of independently constructed plans	**LEARNING OBJECTIVE PROGRESSES TO:** *Develop a complex computer program for a specified task* ***S*ee next page**
INDICATORS	Students can: • specify variables and their data types • independently construct a flexible and robust plan • specify a procedural structure that combines well-chosen actions, conditions and control structures that constitutes a well-structured logical solution to the task • specify a comprehensive set of test cases with expected, boundary and invalid input for testing the program.	Students can: • specify well-chosen scopes for variables, their scopes and data types • specify an indexed data structure • specify a modular structure for the program with well-chosen parameters, including details of the procedural structures of the modules, that constitute a well-structured logical decomposition of the task • specify a comprehensive set of expected, boundary and exceptional input cases for testing the program. • specify variables, constants, and derived values effectively so as to maximise the flexibility and robustness of an independently constructed plan.	
AS	**AS91075 Digital Technologies 1.45** *Construct a plan for a basic computer program for a specified task*	**AS91372 Digital Technologies 2.45** *Construct a plan for an advanced computer program for a specified task*	**AS91637 Digital Technologies 3.46** *Develop a complex computer program for a specified task*
	Level 1 Digital Technologies standards	Level 2 Digital Technologies standards & assessment	Level 3 Technology achievement standards & assessment resources DRAFT

DIGITAL TECHNOLOGIES: CONSTRUCT A SOFTWARE PROGRAM

Construct a software program focuses on constructing a computer program for a specified task including testing and debugging the program to ensure the program works correctly.

Initially students learn to construct basic computer programs in any programming language (drag-and-drop language, specialised programming language, or a general purpose programming language) that include:

- variables, assignment, predefined actions, expressions, and sequence, selection, and iteration control structures; and
- obtains and uses input from a user, sensors, or other external source.

Students progress to constructing complex computer programs using a text based programming language

	LEVEL 6	LEVEL 7	LEVEL 8
LO	*Demonstrate ability to construct a basic software program*	*Demonstrate ability to construct an advanced software program*	*Develop a complex computer program for a specified task*
TEACHER GUIDANCE	To support students to develop an ability to construct a basic software program at level 6, teachers could: • Guide students to independently implement a plan for a basic program in a suitable programming language (drag-and-drop language, specialised programming language, or a general purpose programming language) that uses a procedural structure with well-chosen actions, conditions and control structures that ensures the program is flexible and robust • Guide students on how to set out program code concisely, and document programs with variable names and succinct comments that accurately explain and justify code function and behaviours • Guide students on how to comprehensively test and debug programs in an organised time effective way to ensure that they work on expected, boundary and invalid inputs.	To support students to develop an ability to construct an advanced software program at level 7, teachers could: • Guide students on how to independently implement a plan to construct advanced programs, in suitable programming language, where the modules (including their procedural structures) constitute a well-structured logical decomposition of the tasks • Guide students on how to use variables, constants, and derived values effectively to increase the flexibility and robustness of programs • Guide students on how to set out program code clearly and document programs with variables and module names, and include comments that explain and justify code functions and behaviours • Guide students on how to comprehensively test and debug programs in organised and time effective ways to ensure that programs are correct on expected, boundary's and invalid inputs.	To support students to develop a complex software program, at level 8, teachers could: • Ensure students understand the requirement at this level that the programming language must be a text-based programming language and have an appropriate IDE that includes debugging tools. • Guide students on how to use an Integrated Development Environment (IDE) to develop code following a disciplined development process with cycles of incremental development and testing. • Guide students on how to construct a complex computer program in a text-based programming language that supports object-orientated structuring. • Guide students on how to follow accepted testing and debugging practices using IDE debugging tools to test and debug a program to ensure it works for expected, boundary, and exceptional cases. • Provide opportunities for students to practice using an appropriate IDE to develop code following a disciplined development process with cycles of incremental development and testing. Provide opportunities for students to practice constructing and testing complex computer programs. • Ensure students understand the requirement at this level that a complex program is a program written in a text-based programming language that interacts with a user, includes variables, assignment, predefined actions, expressions, includes sequence, selection, iteration control structures, includes programmer defined methods/ functions/etc with parameters and/or return values, includes calls to the methods/functions/etc, uses structured data, including sequential data (arrays, lists, etc) and compound data (records, objects, tuples etc), uses and updates persistent data in files or databases, has structuring of the methods/functions/etc and data (eg, Classes, modules, encapsulated data structures, packages, etc). • Ensure students have a specified task that requires the development of a complex program to resolve the task. The task can be teacher-given or developed in negotiation with the student. Ensure the task is large enough to justify decomposition of a program into multiple classes (or other high level modules).
INDICATORS	Students can: • write a program with sequence, selection, and iteration control structures • write a program with multiple data types, iteration control structures nested inside other iteration control structures, and structures in which complex logical conditions are expressed economically • comprehensively test and debug the program in an organised and time effective way to ensure the program is correct on all inputs (including expected, exceptional, out-of-range, boundary, and invalid inputs).	Students can: • independently implement a plan for an advanced program in a suitable programming language (preferably a text-based programming language) • construct an advanced program where the modules (including their procedural structures) constitute a well-structured logical decomposition of the task • use variables, constants, and derived values effectively so as to increase the flexibility and robustness of the program • set out the program code clearly and concisely and document the program with comments that explain and justify decisions • comprehensively test and debug the program in an organised and time effective way to ensure the program is correct on expected, boundary and invalid inputs.	Students can: • use an appropriate IDE to develop code and use the IDE debugging tools effectively to identify logic errors and correct a program • follow a disciplined and planned development process with documented cycles of incremental development and comprehensive testing at each cycle to construct a correctly working program • follow accepted debugging practices by interpreting syntax and runtime error messages to identify the underlying errors and correct a program • follow accepted testing and debugging practices for systematically applying test cases and using tracing/debugging statements to identify logic errors and correct a program. • write a computer program in a text-based programming language that includes commented, programmer defined methods/functions/etc with parameters and/or return values, has structuring of the methods/functions/etc and data (eg, classes, modules, encapsulated data structures, packages, etc), and has well-designed algorithmic structures for the individual methods/functions/etc. • write a program that includes functions/methods/procedures that are passed compound data structures (arrays, lists, objects, etc) and modify their contents, and has a well-designed decomposition into functions/methods with well-chosen parameters and has a well-designed structuring of data and methods/functions/etc into classes (or modules, packages, etc) • include explanatory comments and identifiers that support maintainability (including informative comments on functions/methods/procedures) • test their program to ensure it works correctly.
AS	**AS91076 Digital Technologies 1.46** *Construct a basic computer program for a specified task*	**AS91373 Digital Technologies 2.46** *Construct an advanced computer program for a specified task*	**AS91637 Digital Technologies 3.46** *Develop a complex computer program for a specified task*
	Level 1 Digital Technologies standards	Level 2 Digital Technologies standards & assessment	Level 3 Technology achievement standards & assessment resources DRAFT

DIGITAL TECHNOLOGIES: KNOWLEDGE OF ELECTRONIC ENVIRONMENTS

Knowledge of electronic environments focuses on the concepts and operational function of components that underpin the understanding of how electronic environments (functional combinations of hardware and embedded software in the real world, ie, circuits, prototypes or products) are developed, assembled and tested.

Initially students learn about basic components and the concepts that describe the behaviour of a circuit. Students progress from this to more advanced understanding of circuit and embedded programming concepts and learn about an increasing range of components and their operation function in real circuits. At the highest level, students will be able to discuss complex electronic environments in terms of their subsystems and programming structures and apply some basic mathematical calculations within this discussion.

	LEVEL 6	LEVEL 7	LEVEL 8
LO	*Demonstrate understanding of basic concepts and components in electronic environments*	*Demonstrate understanding of advanced concepts and components in electronic environments*	*Demonstrate understanding of complex concepts and components in electronic environments*
TEACHER GUIDANCE	To support students to develop understandings about basic concepts and components in electronic environments at level 6, teachers could: • Provide the opportunity for students to learn about basic concepts through practical settings, for example, test conductors, insulators and semiconductor diode using a multimeter (ohms) or a light bulb and battery or learn why a circuit must be complete by identifying hidden breaks in a circuit using a multimeter. • Guide students to identify basic components and their symbols Support students to experiment with basic components in simple circuits to consolidate their understanding • Guide students to classify a provided selection of components in a tray as sensors, actuators or processors. • Provide opportunity for discussion about the components properties in terms of energy transfer, for example, an LDR converting light to electrical energy and an LED converting electrical energy to light • Support students to use symbols to create schematics for simple circuits, such as a simple microcontroller circuit with at least one input and a few simple outputs • Provide a range of practical experiences, for example: exploring the properties of series and parallel connections using LEDs in a circuit; using a multimeter in a simple LED-resistor circuit to introduce the concept of voltage as an energy level, and the concepts of current and resistance. • Provide opportunity for students to program a simple microcontroller to perform basic functions such as blinking LEDs controlled by a single switch. Students will be provided with the basic program structures for this. • Guide students to examine simple two-loop circuits, including those with a microcontroller to identify and describe voltage divider and transistor switch subsystems in these.	To support students to develop understandings about advanced concepts and components in electronic environments at level 7, teachers could: • Provide opportunity for students to learn about advanced concepts, including power and heat dissipation, analogue and digital signals, amplification, logical AND/OR and truth tables, parallel and series, how a single component type may have varied roles through hands-on practical work and research etc. • Provide opportunities for students to discuss and investigate practically, software programme development using advanced concepts, such as variables, binary notation (bits, bytes and words), logical structuring of software programmes (eg, flowcharting) and the use of subroutines and variables. • Provide opportunity for students to experiment with an extended range of components in circuits – such as a diode (pn and zener), capacitor (various types), npn transistor or FET – and an extended range of common sensors and actuators, such as Hall sensor, servo etc. • Guide students to explore the properties of integrated devices, for example, H-bridge, voltage regulators. • Guide students to research information (books,online etc) about the properties and operation of components and guide them to in selecting relevant material from these sources. • Support students to perform calculations, including power rating, parallel and series, based on parameters important in the behaviour of real circuits. • Provide the opportunity for students to explore an extended set of subsystems – including temperature sensors, LCDs, amplifier stages etc – and enable students to recognise these in advanced circuit schematics.	To support students to develop understandings about complex concepts and components in electronic environments at level 8, teachers could: • Provide opportunity for students to learn about complex hardware concepts, for example, IR and radio transmitting subsystems, amplifying stages, noise reduction circuits, UART, bus subsystems, through hands-on practical work and internet research, etc. • Provide opportunity for students to learn about complex software concepts, for example, variables, binary notation (bits, bytes and words), protocols (I2C, RS232), macros, flags, interrupts, counters, XOR, bitwise AND/OR, pwm, through hands-on practical work and internet research, etc. • Provide opportunity for students to learn about complex components – such as FETs, npn and pnp transistors, voltage regulators, SCRs, gates, H-bridges, op-amps, data latches, keypads, LCD and other displays, pressure and proximity sensors, servo and stepper motors etc – and describe these in terms of their operational function in different contexts. • Provide opportunity for students to identify, describe and explain some complex subsystems in circuits, such as transistor combinations (eg, push-pull), transistor configurations (eg, common emitter), extended gate arrangements, power supply circuits, FET circuits. • Provide opportunity for students to learn about software programme development through the logical structuring of software embedded programmes (eg, flowcharting/state machines) and the use of subroutines and variables. • Support students to choose an embedded platform based upon its features, for example, a selection between various PICAXE, Microchip or AVR microcontrollers, or Arduino, Raspberry Pi/otr Tablet platform etc. • Guide students to research information (books, online etc) about the properties and operation of components and ensure they are able to determine relevant material and critique and/or synthesise this in ways that support their understanding. • Support opportunities for students to perform complex calculations – such as gain, RMS and power values – based on parameters important in the behaviour of real circuits.
INDICATORS	Students can: • analyse basic concepts of electronics to explain the behaviour of electronic systems • discuss the operational function of electronic components in a practical context.	Students can: • use advanced concepts of electronics to discuss the implications of multiple variables on the performance of electronic environments • discuss the advantages and disadvantages of different electronic components to achieve desired advanced operational functions.	Students can: • discuss complex software concepts • discuss complex hardware concepts.
AS	**AS91077 Digital Technologies 1.47** *Demonstrate understanding of basic concepts used in the design and construction of electronic environments*	**AS91374 Digital Technologies 2.47** *Demonstrate understanding of advanced concepts used in the construction of electronic environments*	**AS91638 Digital Technologies 3.47** *Demonstrate understanding of complex concepts used in the design and construction of electronic environments*
	Level 1 Digital Technologies standards & assessment resources	Level 2 Digital Technologies standards & assessment resources	Level 3 Technology achievement standards & assessment resources DRAFT

DIGITAL TECHNOLOGIES: DEVELOP AN ELECTRONIC ENVIRONMENT

Development of electronic environments focuses on the analysis of how electronic environments (functional combinations of hardware and embedded software in the real world i.e. circuits, prototypes or products) work in terms of their components, subsystems and software and how these components may be selected, subsystems put together and the hardware and software tested and debugged so that the electronic environment is functional with respect to agreed specifications. The model produced through these skills is a necessary precursor to developing a functional electronic and embedded system.
Initially, students learn basic functional modelling, circuit analysis, subsystem assembly and adjustment, testing and debugging skills. Students progress from this to more advanced skills to deal with more advanced and eventually complex environments. This progression will involve the introduction of more complex calculation and competency in the use and interpretation of data from devices such as multimeters (extended function), oscilloscopes and other test instruments. At the highest level, students will be able to analyse and develop complex electronic environments in terms of their subsystems and programming structures and employ mathematical calculations as part of this process.

	LEVEL 6	LEVEL 7	LEVEL 8
LO	*Demonstrate ability to develop a basic electronic environment*	*Demonstrate ability to develop an advanced electronic environment*	*Demonstrate ability to develop a complex electronic environment*
TEACHER GUIDANCE	To support students to demonstrate ability to develop a basic electronic environment at level 6, teachers could: • Provide, or develop in negotiation with the student, specifications for an electronic environment that will require applying some basic interfacing procedures. • Provide experiences that involve functional modelling eg, breadboard simple circuits, use of simple simulation, kit-sets. • Support students to analyse basic circuits (including those with a microcontroller) in terms of their subsystems (eg, voltage divider and transistor switch). • Guide students to use functional modelling to perform measurements in, and to test, debug and make adjustments to voltage divider and transistor switch subsystems. • Guide students to use functional modelling to develop software for a simple embedded system, given provided programme structures. • Guide students to use functional modelling to test, debug and make adjustments to simple embedded software. • Guide students to use functional modelling to interface subsystems to each other and to the embedded software in a microcontroller. • Provide and explain a datasheet for a common component (eg, a resistor or LED). • Provide opportunity for students to practice simple calculations based on selecting components for real circuits (eg, the value of a resistor to match an LED). • Guide students to test and debug an integrated electronic environment to ensure functionality.	To support students to demonstrate ability to develop an advanced electronic environment at level 7, teachers could: • Provide, or develop in negotiation with the student, specifications for an electronic environment that will require applying some advanced interfacing procedures. • Support students to analyse advanced circuits (those involving microcontrollers, timer/counter, amplifier stages etc) in terms of their subsystems. • Provide functional modelling tools to enable students to perform measurements in, and to test, debug and make adjustments to advanced circuit subsystems. • Guide students to use functional modelling to develop clearly annotated software (including variables, subroutines and an extended range of commands) for an advanced embedded system. • Provide functional modelling tools to enable students to test, debug and make adjustments to advanced embedded software. • Guide students to use functional modelling to interface subsystems to each other and to the embedded software in a microcontroller or other development environment such as Arduino, Raspberry Pi or tablet interface. • Provide opportunity for students to practise interpreting datasheets and undertaking calculations based on real circuits, including voltage, current and power (eg, selecting components for a motor driver subsystem based on average and maximum power requirements). • Support students to test and debug an integrated electronic environment to ensure functionality.	To support students to demonstrate ability to develop a complex electronic environment at level 8, teachers could: • Provide, or develop in negotiation with the student, specifications for an electronic environment that will require applying some complex interfacing procedures. • Support students to analyse complex circuits (those involving FETs, npn and pnp transistors, SCRs, op-amps, LCD displays, servo and stepper motors etc.) in terms of their subsystems. • Provide functional modelling tools to enable students to perform measurements in, and to test, debug, and make adjustments to complex circuit subsystems. • Guide students to use functional modelling to develop clearly annotated, well-structured software (including communication protocols, macros, flags, interrupts, counters, bitwise AND/OR, pwm) for a complex embedded system. • Provide functional modelling tools to enable students to test, debug and make adjustments to complex embedded software. • Guide students to use functional modelling to interface subsystems to each other and to the embedded software in a microcontroller or other development environment. • Guide students to perform complex calculations, including gain, RMS values and power, based on parameters important in the behaviour of real circuits. • Support students to interpret datasheets and undertake calculations based on real circuits (eg, selecting component values and subsystem design so that circuits and data are matched to each other). • Ensure students are able to test and debug the integrated electronic environment to ensure functionality.
INDICATORS	Students can: • use datasheets or calculations to assist in choosing appropriate component types and values for the sensor and actuator subsystems • write well-structured, clearly annotated, readily understandable interface software given simple programme structures • interface subsystems to each other and to the embedded software in a microcontroller • test and debug a functional model of the interface.	Students can: • use provided functional sensor subsystems to interact with the environment • use provided functional actuator subsystems to interact with the environment • modify sensor subsystems to substantially improve the quality of the data delivered by the interface • modify actuator subsystems to substantially improve the way they work • write well-structured, clearly annotated, readily understandable software that interfaces effectively with the data provided by the sensors and with the actuators it controls • interface subsystems to each other and to the embedded software in a microcontroller • test and debug a functional model of the interface to achieve and demonstrate substantially improved operation.	Students can: • devise and apply functional sensor subsystems that interact with the environment • devise and apply functional actuator subsystems to interact with the environment • analyse and modify sensor subsystems to substantially improve the quality of the data delivered by the interface • analyse and modify actuator subsystems to substantially improve the way they work • write well-structured, clearly annotated, readily understandable embedded software that interfaces effectively with the data provided by the sensors and with the actuators it controls • interface subsystems to each other and to the embedded software in a microcontroller or other development environment. • analyse, modify, test and debug a functional model of the interface to achieve and demonstrate substantially improved operation.
AS	**AS91078 Digital Technologies 1.48** Implement basic interfacing procedures in a specified electronic environment	**AS91375 Digital Technologies 2.48** Implement advanced interfacing procedures in a specified electronic environment	**AS91639 Digital Technologies 3.48** Implement complex interfacing procedures in a specified electronic environment
	Level 1 Digital Technologies standards & assessment resources	**Level 2 Digital Technologies standards & assessment resources**	**Level 3 Digital Technologies standards & assessment resources**

DIGITAL TECHNOLOGIES: ASSEMBLE AND TEST ELECTRONIC AND EMBEDDED SYSTEMS

The assembly and testing of electronic and embedded system is focused on developing the skills needed to integrate technologies (hardware, software, mechanical) to produce a working prototype. These skills follow directly from those acquired during the development of an electronic environment as a functional model. It is also about the application of testing, debugging and modification skills to ensure the prototype is operational, fit for purpose and meets specifications.

Initially students learn basic assembly and testing skills and about working safely in the classroom and/or workshop environment. Students progress from here to levels that require more advanced and complex skills. This progression may require developing competency in calculating values and in the use and interpretation of data from devices such as multimeters (extended functions), oscilloscopes and other test instruments. At the highest level, students will be able to use complex techniques to construct and debug electronic and embedded systems to meet design specifications.

	LEVEL 6	LEVEL 7	LEVEL 8
LO	*Demonstrate basic assembly and testing techniques used in electronic and embedded systems*	*Demonstrate advanced assembly and testing techniques used in electronic and embedded systems*	*Demonstrate complex assembly and testing techniques used in complex electronic and embedded systems*
TEACHER GUIDANCE	To support students to demonstrate basic assembly and testing techniques used in electronic and embedded systems at level 6, teachers could: • Provide, or develop in negotiation with the student, specifications for an electronic environment that will require basic techniques. • Provide opportunity for students to select components that match a given schematic. • Provide instruction for students in the design and production of a simple PCB (printed circuit board) using 'pen and etch' technique. • Provide opportunity for students to develop correct soldering techniques. • Ensure students apply acceptable standards of cleanliness in their work area and care for their equipment. • Provide opportunity to discuss necessary safety procedures for soldering, drilling and PCB production. • Provide opportunity for students to develop skills in drilling, populating and soldering up of a circuit on a PCB. • Provide opportunity for students to work with other circuit platforms, such as a copper tape and veroboard. • Provide opportunity for students to develop the art of visual critical inspection of their handiwork, including all hardware (circuits and mechanicals) as well as software programs. • Ensure students test each soldered joint for continuity with a multimeter, as each joint is completed. • Guide students to use a multimeter to test components (eg, resistor values) and locate basic faults in a real circuit, such as a bad joint, by measuring voltage levels at different points. • Provide opportunity for students to develop, test and download programs into embedded software, given supplied programme structures as a starting point. • Provide opportunity for students to inspect and debug software programs. • Guide students to employ basic techniques to evaluate, test and debug the assembled electronic and embedded system so that the overall system is functional. • Provide students with opportunity to work and cooperate in groups. • Ensure students understand how an outcome is evaluated against specifications in a brief.	To support students to demonstrate advanced assembly and testing techniques used in electronic and embedded systems at level 7, teachers could: • Provide, or develop in negotiation with the student, specifications for an electronic environment that will require advanced techniques. The environment will include more than one subsystem and include at least one multi-pin device. • Provide opportunity for students to select an extended range of components to match a schematic. • Provide instruction for students in the design and production of a PCB using CAD techniques Provide opportunity for students to develop advanced soldering techniques (eg, temperature controlled, desoldering etc.) so that students can achieve consistently reliable results. • Provide opportunity for students to use advanced multimeter functions to test an extended range of components (eg, capacitor values) to locate faulty components and other problems in a circuit, visual inspection, using an extended range of techniques in a logical manner (eg, voltage levels at the system and progressive subsystem levels). • Provide opportunity for students to perform systematic and logical testing, evaluation of data and debugging in the electronic environment. • Provide instruction in and examples for students that show how calculation and measurement can assist in the testing and debugging of the hardware and software in the system. • Provide instruction in and examples of advanced techniques for the development, testing and debugging of clearly annotated embedded software that uses features such as variables and subroutines and simple data structures such as an array. • Guide students to employ advanced techniques to evaluate, test and debug the assembled electronic and embedded system so that the overall system is functional.	To support students to demonstrate complex assembly and testing techniques used in electronic and embedded systems at level 8, teachers could: • Provide, or develop in negotiation with the student, specifications for an electronic environment that will require complex techniques. The environment will include several subsystems and include at least three multi-pin devices. • Provide opportunity for students to select an extended range of components to match a schematic. • Guide students in the design and production of a quality, complex PCB to a high standard, using techniques that will achieve this result. • Provide opportunity for students to develop complex soldering techniques so that students can achieve consistently reliable results. • Provide opportunity for students to employ complex techniques to assemble a functional, reliable and well-laid out hardware platform on PCB(s) (organised layout, component size considerations, component and user safety, off-board connections, vias, logical routing, siting and protection of off-board components, with easy access to testing points. • Provide opportunity for students to acquire complex programming skills for the development, testing and debugging of clearly annotated embedded software that uses features such as communication protocols, macros, flags, interrupts and counters. • Provide opportunity for students to use complex diagnostic techniques (advanced multimeter, oscilloscope, signal generator, logic tester/analyser etc.) functions to test an extended range of component functions in situ, including integrated circuits. • Guide students to perform systematic and logical testing, evaluation of data and debugging in the electronic environment. • Guide students to employ calculation and measurement in the process of testing and debugging of the hardware and software in the system. • Support students to employ complex techniques to evaluate, test and debug the assembled electronic and embedded system so that the overall system is functional.
INDICATORS	Students can: • construct and test reliable functional systems with well-soldered joints; suitable track and component layout and secure, reliable, well-organised connections to any components that are mounted off the board • write and debug embedded software so that the program is logical, efficient and clearly annotated.	Students can: • develop and produce a printed circuit board (PCB) using PCB CAD software • construct and test reliable functional circuits on PCB, with improved track layout and soldering • write and debug well-structured, clearly annotated, and readily understandable embedded software which uses one extended feature or specialised command.	Students can: • use PCB (printed circuit board) CAD software to develop a PCB layout that will preserve signal integrity • construct, test, analyse and modify reliable functional circuits on PCB, with substantially improved track layout and soldering • write, debug and modify well-structured, clearly annotated, and readily understandable embedded software • analyse and effectively manage signal and data parameters.
AS	**AS91079 Digital Technologies 1.49** *Implement basic techniques in constructing a specified electronic and embedded system*	**AS91376 Digital Technologies 2.49** *Implement advanced techniques in constructing a specified electronic and embedded system*	**AS91640 Digital Technologies 3.49** *Implement complex techniques in constructing a specified complex electronic and embedded system*
	Level 1 Digital Technologies standards & assessment resources	Level 2 Digital Technologies standards & assessment resources	Level 3 Technology achievement standards & assessment resources DRAFT

DIGITAL TECHNOLOGIES: KNOWLEDGE OF DIGITAL INFRASTRUCTURE

Knowledge of digital infrastructure focuses on the concepts of digital infrastructure within personal computers, local area networks (LANs) and Wide Area Networks (WANs).

Initially students learn about the common components of basic digital infrastructures consisting of personal computer hardware, associated peripherals and system software. Students learn about the purpose of the components, typical connections and data flow between components, characteristics of components that limit their inter-operability, and procedures and protocols for installing or replacing a component or a program. Student's progress to learn about complex concepts of digital infrastructure associated with LANs and WANs.

	LEVEL 6	LEVEL 7	LEVEL 8
LO	*Demonstrate understanding of digital infrastructure components*	*Demonstrate understanding of LAN infrastructure systems*	*Demonstrate understanding of WAN infrastructure systems*
TEACHER GUIDANCE	To support students to develop understandings about of digital infrastructure components at level 6, teachers could: • Provide students with the opportunity to explore digital infrastructure in order to identify personal computer hardware, associated peripherals and system software. • Provide the opportunity for students to explore the purpose of components and their characteristics. • Assist in the refinement of reflective and inquiry questions related to the understanding of procedures and protocols associated with basic infrastructure. • Guide students on how to prepare reports including ways to structure a report and literacy strategies to support report writing in a way that will allow students to describe, explain, and discuss. • Provide opportunities for students to practice report writing.	To support students to develop understandings about LAN infrastructure systems at level 7, teachers could: • Provide students with the opportunity to explore local area networks (LAN s) consisting of a number of networked devices which includes at least three PCs connected with an unmanaged switch, simple server elements and a single connection to the internet. • Provide the opportunity for students to explore the characteristics and purposes of LANs, and discuss their components, network layers, bandwidth, data transmission modes, IP addressing, DHCP (Dynamic Host Configuration Protocol), NAT (Network Address Translation) and ICMP (Internet Control Message Protocol). • Assist in the refinement of reflective and inquiry questions related to the understanding of procedures and protocols associated with the development and maintenance of LANs. • Support students to prepare reports including ways to structure a report and literacy strategies to support report writing in a way that will allow students to describe, explain, and discuss. • Ensure students have opportunities to practice report writing.	To support students to develop understandings about WAN infrastructure systems at level 8, teachers could: • Provide students with the opportunity to explore wide area networks (WANs). • Provide the opportunity for students to explore the characteristics and purposes of WANs, and discuss their components, and the layers in the TCP/IP networking model. • Assisting in the refinement of reflective and inquiry questions related to the understanding of procedures and protocols associated with the development and maintenance of WANs. • Support students to prepare reports including ways to structure a report and literacy strategies to support report writing in a way that will allow students to describe, explain, and discuss. • Ensure students have opportunities to practice report writing.
INDICATORS	Students can: • describe and identify the purpose of the components of basic digital infrastructures • describe the typical connections and data flow between components of a basic digital infrastructure • describe the key characteristics of components of a basic digital infrastructure that limit their inter-operability • describe a procedure or protocol for installing or replacing a physical component or a program • explain how the purpose of components determines the connections between components and the typical flow of data along them • explain how the key characteristics of components limit their inter-operability • explain the importance of procedures and protocols when installing or replacing a component or a program • discuss the characteristics and limitations of the connections that carry data between components • discuss the key characteristics used to specify each kind of component in terms of inter-operability, tradeoffs, efficiencies, cost, and context of use.	Students can: • describe networking concepts such as the characteristics and purposes of a local area network (LAN), standard networking models, bandwidth, data transmission modes, IP addressing, DHCP, NAT, and ICMP • explain why the components have been used in a LAN to achieve the desired characteristics • explain how the connection technologies allow the components to function in a LAN • describe the access control method used in the Ethernet architecture. • compare and contrast the characteristics and the purposes of peer-to-peer LANs and client/server LANs • explain the layers in the TCP/IP networking model and the role of this model in a LAN architecture • explain IP (Internet Protocol) addressing with reference to static addresses and dynamically obtained addresses • discuss the advantages and disadvantages of the common cable, fibre and wireless technologies for connecting the components of a LAN • discuss IP addressing schema including the consequences for static addresses and dynamically obtained addresses • discuss how the access control method used in the Ethernet architecture manages Ethernet traffic on a LAN.	Students can: • describe Wide Area Network (WAN) technologies such as WAN protocols, basic routing principles including static routing, common wired, optical and wireless technologies, WAN architecture specified in terms of physical topologies and logical topologies. • explain why the components have been used in a WAN to achieve the desired characteristics • describe the use of WAN protocols in a WAN architecture. • compare and contrast the characteristics and the purposes of different WAN technologies and components • explain the layers in the Transmission Control Protocol (TCP)/ Internet Protocol (IP) networking model and the role of this model in a WAN architecture • discuss IP (Internet Protocol) addressing with reference to static addresses and dynamically obtained addresses. • explain the WAN technologies and components of a WAN • explain IP addressing schema • explain how NAT (Network Address Translation) provides a form of firewall • explain the management procedures for a WAN link.
AS	**AS910780 Digital Technologies 1.50** *Demonstrate understanding of the common components of basic digital infrastructures*	**AS91377 Digital Technologies 2.50** *Demonstrate understanding of local area network technologies*	**AS91641 Digital Technologies 3.50** *Demonstrate understanding of wide area network technologies*
	Level 1 Digital Technologies standards & assessment resources	Level 2 Digital Technologies standards & assessment resources	Level 3 Technology achievement standards & assessment DRAFT

DIGITAL TECHNOLOGIES: DESIGN A DIGITAL INFRASTRUCTURE SYSTEM

Design a digital infrastructure system refers to the assembly and management of a specified system.

Designing a digital infrastructure system requires particular techniques to be used to select, assemble, configure and install components for a specified purpose. Components are hardware and software.

When designing an infrastructure system students will employ standard procedures for installing and configuring hardware and peripherals (eg, systematic use of procedures specified in manufacturer OEM manuals, antistatic procedures, electrical safety procedures, and relevant OSH regulations) and standard procedures for installing and configuring software (eg, standard best practice and procedures specified in manuals, installation guides, installation programs, and system documentation including Help facilities).

Initially students learn to assemble and service a personal computer system. Students progress to assembling, configuring and managing local and wide area networks.

	LEVEL 6	LEVEL 7	LEVEL 8
LO	*Select components for a single computer system for a specified purpose*	*Select components and parameters for a local area network for a specified purpose*	*Select components and topology for a wide area network for a specified purpose*
TEACHER GUIDANCE	To support students to develop skills in constructing a single computer system at level 6, teachers could: • Ensure students can identify components of a single computer system. • Provide an opportunity for students to select components [eg, memory capacity, graphics cards, peripherals] that have sufficient capacity to meet the needs of specified software. • Guide students to identify and resolve installation and configuration faults related to hardware and software.	To support students to develop skills in constructing and managing a local area network (LAN) at level 7, teachers could: • Ensure students can identify components of a LAN. • Ensure students understand and follow administrative procedures to manage a LAN. • Provide students with an opportunity to undertake testing procedures through the use of: hardware identification tools, system software identification tools and stand alone programs. • Provide an opportunity for students to select network components [eg, LAN speed, routers/switches] that have sufficient capacity for a specified purpose [eg, file sharing, file storage, shared services]. • Guide students to identify and resolve installation and configuration faults related to hardware, network architecture and software.	To support students to develop skills in constructing and managing a wide area network (WAN) at level 8, teachers could: • Ensure students can identify components of a WAN. • Ensure students understand and follow administrative procedures to manage a WAN. • Provide an opportunity for students to estimate the capacity of a design for a network of computers and their access to a Wide Area Network [eg, estimate how many simultaneous streaming video feeds a network could support, or estimate the number of minutes of music that could be stored on a file system]. • Guide students to identify and resolve installation and configuration faults related to hardware and network architecture.
INDICATORS	Students can: • show accuracy and independence in following standard procedures for installing and configuring hardware, software and peripherals of a personal computer system • show accuracy and independence in diagnosing and troubleshooting to identify and resolve given installation and configuration faults in a system. • diagnose and troubleshooting to identify and resolve given installation and configuration faults in a system, in a manner that is economical in time, effort and resources.	Students can: • follow standard administrative procedures to manage a LAN, showing organisation, confidence, accuracy and independence • diagnose and troubleshooting a LAN to identify and resolve given installation and configuration faults in a manner that is economical in time, effort and resources showing organisation, confidence, accuracy and independence.	Students can: • follow standard administrative procedures to manage a WAN, showing organisation, accuracy and independence • diagnose and troubleshooting a WAN to identify and resolve given installation and configuration faults in a manner that is economical in time, effort and resources, and shows organisation, accuracy and independence.
AS	**AS91073 Digital Technologies 1.51** *Implement basic procedures servicing a personal computer system*	**As91370 Digital Technologies 2.51** *Implement procedures for administering a local area network*	**AS91642 Digital Technologies 3.51** *Implement procedures for administering a wide area network*
	Level 1 Digital Technologies standards & assessment resources	Level 2 Digital Technologies standards & assessment resources	Level 3 Technology achievement standards & assessment resources DRAFT