Digital Technologies

2019 Subject Outline

Stage 2

This Board-accredited Stage 2 subject outline  
will be taught from **2019.**

Published by the SACE Board of South Australia,  
60 Greenhill Road, Wayville, South Australia 5034

Copyright © SACE Board of South Australia 2017

First published 2017

Published online November 2017  
ISBN 978 1 74102 840 9 (online Microsoft Word version)

ref:

*This subject outline is accredited for teaching at Stage 2 from 2019*

contents

Introduction 1

Subject description 1

Capabilities 2

Aboriginal and Torres Strait Islander knowledge, cultures, and perspectives 5

Learning scope and requirements 6

Learning requirements 6

Content 6

Assessment scope and requirements 19

Evidence of learning 19

Assessment design criteria 19

School assessment 20

External assessment 22

Performance standards 22

Assessment integrity 25

Support materials 26

Subject-specific advice 26

Advice on ethical study and research 26

Introduction

Subject description

Digital Technologies is a 10-credit subject or a 20-credit subject at Stage 1, and a 20‑credit subject at Stage 2.

Digital technologies have changed the ways that people think, work, and live. The application of digital technologies can lead to discoveries, new learning, and innovative approaches to understanding and solving problems.

The study of Digital Technologies provides a platform for deep interdisciplinary learning. Students make connections with innovation in other fields and across other learning areas.

In Digital Technologies students create practical, innovative solutions to problems of interest. By extracting, interpreting, and modelling real-world data sets, students identify trends and examine sustainable solutions to problems in, for example, business, industry, the environment, and the community. They investigate how potential solutions are influenced by current and projected social, economic, environmental, scientific, and ethical considerations, including relevance, originality, appropriateness, and sustainability.

Innovation in Digital Technologies involves students creating new ways of doing things, generating their own ideas and creating digital solutions to problems of interest. Solutions may take the form of a product, prototype, and/or proof of concept. Students are encouraged to experiment and learn from what does not work as planned, as well as from what does work. Innovation may also include students designing solutions that improve existing processes or products.

Students use computational thinking skills and strategies to identify, deconstruct, and solve problems that are of interest to them. They analyse and evaluate data, test hypotheses, make decisions based on evidence, and create solutions. Through the study of Digital Technologies, students are encouraged to take ownership of problems and design, code, validate, and evaluate their solutions. In doing so, they develop and extend their understanding of designing and programming, including the basic constructs involved in coding, array processing, and modularisation.

At Stage 2, students develop and apply their skills in computational thinking and in program design, and engage in iterative project development, where a product or prototype is designed and tested and/or implemented in stages. They follow agile practices and/or iterative engineering design processes. Learning environments in Digital Technologies may include physical, online, and/or simulated spaces.

Digital Technologies promotes learning through initiative, collaboration, creativity, and communication, using project- and inquiry-based approaches.

Capabilities

The capabilities connect student learning within and across subjects in a range of contexts.

The SACE identifies seven capabilities. They are:

* literacy
* numeracy
* information and communication technology (ICT) capability
* critical and creative thinking
* personal and social capability
* ethical understanding
* intercultural understanding.

Literacy

In this subject students extend and apply their literacy capability by, for example:

* using digital and other communication strategies to engage in collaborative projects and tasks
* interpreting, analysing, and evaluating primary and secondary data sets relevant to problems of interest
* using a range of visualisation tools to identify and deconstruct problems and design solutions (product, prototype, and/or proof of concept), incorporating the terminology and conventions of digital technologies
* becoming competent and confident designers of the features of digital solutions
* using accurate and appropriate terminology to interpret, discuss, and explain concepts, issues, problems, and solutions involving digital technologies
* communicating project ideas using tools such as storyboards and other visual representations
* reading and interpreting online documentation and tutorial materials that support coding.

Numeracy

In this subject students extend and apply their numeracy capability by, for example:

* using computational thinking in identifying, deconstructing, and solving problems of interest
* predicting trends by analysing data sets
* displaying numerical and statistical data in appropriate formats, using visualisation tools
* examining the usefulness of results and preparing validation plans for calculating outputs of digital solutions
* using code that enables manipulation of numerical data in digital solutions
* applying appropriate mathematical and logical concepts and thinking in programming.

Information and communication technology (ICT) capability

In this subject students extend and apply their ICT capability by, for example:

* locating and accessing information using digital technologies
* using a range of digital technologies to extract, interpret, and model data sets
* presenting findings using multimodal approaches, including infographics
* using computational thinking skills and strategies to identify, deconstruct, and solve problems of interest
* using program-design skills to develop digital solutions (product, prototype, and/or proof of concept)
* using digital technologies to work collaboratively on various projects, including creating innovative solutions
* understanding the impact of ICT, its development and application, and how it shapes and is shaped by social agendas.

Critical and creative thinking

In this subject students extend and apply their critical and creative thinking capability by, for example:

* visualising possibilities and scoping solutions
* using computational thinking skills to analyse and evaluate data, and make predictions and decisions
* identifying and deconstructing problems of interest
* using abstraction to deconstruct problems
* exploring and testing hypotheses using design-related investigations
* using initiative in designing projects and solutions
* critically evaluating existing and proposed digital solutions
* thinking creatively about ways of working collaboratively to design digital solutions
* designing innovative, creative, and appropriate digital solutions using agile practices
* applying programming as a creative tool.

Personal and social capability

In this subject students extend and apply their personal and social capability by, for example:

* working collaboratively to identify team roles, protocols, and deliverables
* collaborating with stakeholders to create and evaluate innovative digital solutions
* making decisions and taking initiative in designing projects
* exploring innovation and developing entrepreneurial skills
* sharing and discussing ideas about problems, progress, and innovative solutions
* listening to and respecting the perspectives of others
* acquiring practical skills, knowledge, and understanding related to the design, development, and use of digital solutions
* forming a critical understanding of how digital solutions affect individuals, groups, and/or societies
* planning effectively and managing time.

Ethical understanding

In this subject students extend and apply their ethical understanding capability by, for example:

* considering the ethical implications of digital solutions on individuals, groups, and/or societies now and in the future
* considering ethical strategies for working with individuals and groups
* evaluating the ethical issues surrounding the collection, storage, and use of data
* making ethical decisions based on evidence
* using data and reporting the outcomes accurately and fairly
* recognising the importance of responsible participation in social, economic, environmental, scientific, and/or ethical decision-making
* increasing critical understanding of the appropriate and ethical uses of digital technology
* understanding the implications of data use for personal, social, and ethical participation in the wider community
* evaluating the reliability of information for accurate decision-making.

Intercultural understanding

In this subject students extend and apply their intercultural understanding capability by, for example:

* exploring design issues in local, national, and global contexts to expand knowledge of, and create solutions for, a diverse range of individuals, groups, and societies
* respecting and engaging with different cultural views and customs, and exploring these interactions using digital technologies
* recognising that engaging with different perspectives enhances own knowledge, understanding, and solutions
* being open-minded and receptive to changes in design and project development
* understanding that the progress of designing and implementing a digital solution influences and is influenced by cultural factors.

Aboriginal and Torres Strait Islander knowledge, cultures, and perspectives

In partnership with Aboriginal and Torres Strait Islander communities, and schools and school sectors, the SACE Board of South Australia supports the development of high-quality learning and assessment design that respects the diverse knowledge, cultures, and perspectives of Indigenous Australians.

The SACE Board encourages teachers to include Aboriginal and Torres Strait Islander knowledge and perspectives in the design, delivery, and assessment of teaching and learning programs by:

* providing opportunities in SACE subjects for students to learn about Aboriginal and Torres Strait Islander histories, cultures, and contemporary experiences
* recognising and respecting the significant contribution of Aboriginal and Torres Strait Islander peoples to Australian society
* drawing students’ attention to the value of Aboriginal and Torres Strait Islander knowledge and perspectives from the past and the present
* promoting the use of culturally appropriate protocols when engaging with and learning from Aboriginal and Torres Strait Islander peoples and communities.

Learning scope and requirements

Learning requirements

The learning requirements summarise the knowledge, skills, and understanding that students are expected to develop and demonstrate through their learning in Stage 2 Digital Technologies.

In this subject, students are expected to:

1. apply computational thinking skills, including abstraction, to approach, identify, deconstruct, and solve problems of interest

2. analyse data sets related to problems of interest to identify patterns and/or trends, draw conclusions, and make predictions

3. apply iterative project-development techniques to manage and evaluate proposed digital solutions to problems of interest

4. apply design and programming skills to create and document digital solutions

5. research and discuss ethical considerations in digital technologies

6. work individually and collaboratively to create and explain digital solutions.

Content

Stage 2 Digital Technologies is a 20-credit subject that consists of the following focus areas:

* Focus area 1: Computational thinking
* Focus area 2: Design and programming
* Focus area 3: Data analytics
* Focus area 4: Iterative project development.

Students study all four focus areas.

The focus areas are not intended to be taught independently. They are not necessarily designed to be of equivalent length—teachers may allocate more time to some than to others—and they may be sequenced and structured to suit individual cohorts of students.

Computational thinking underpins the learning in this subject. In applying their computational thinking skills, students apply logical steps to identify and deconstruct problems that are of interest to them, recognise patterns through abstraction, design algorithms, and create innovative digital solutions.

(Note: a digital solution may be implemented by the student or may be a working prototype or proof of concept.)

Computational thinking skills are applied in each focus area.

In Focus area 1: Computational thinking, students develop and extend their computational thinking skills and strategies to identify, deconstruct, and solve problems of interest. These strategies include pattern recognition, abstraction, and algorithm design. In Focus area 2: Design and programming, students analyse a problem, and design, write the code for, test, and implement a solution. In Focus area 3: Data analytics, students analyse data sets in order to understand a problem, test a hypothesis, and draw conclusions from which to make decisions. In Focus area 4: Iterative project development, students scope problems, plan a project, clarify project features, and develop and evaluate appropriate code.

Pivotal to student learning is the development of the capabilities and, in particular, the skills for creativity, collaboration, and innovation.

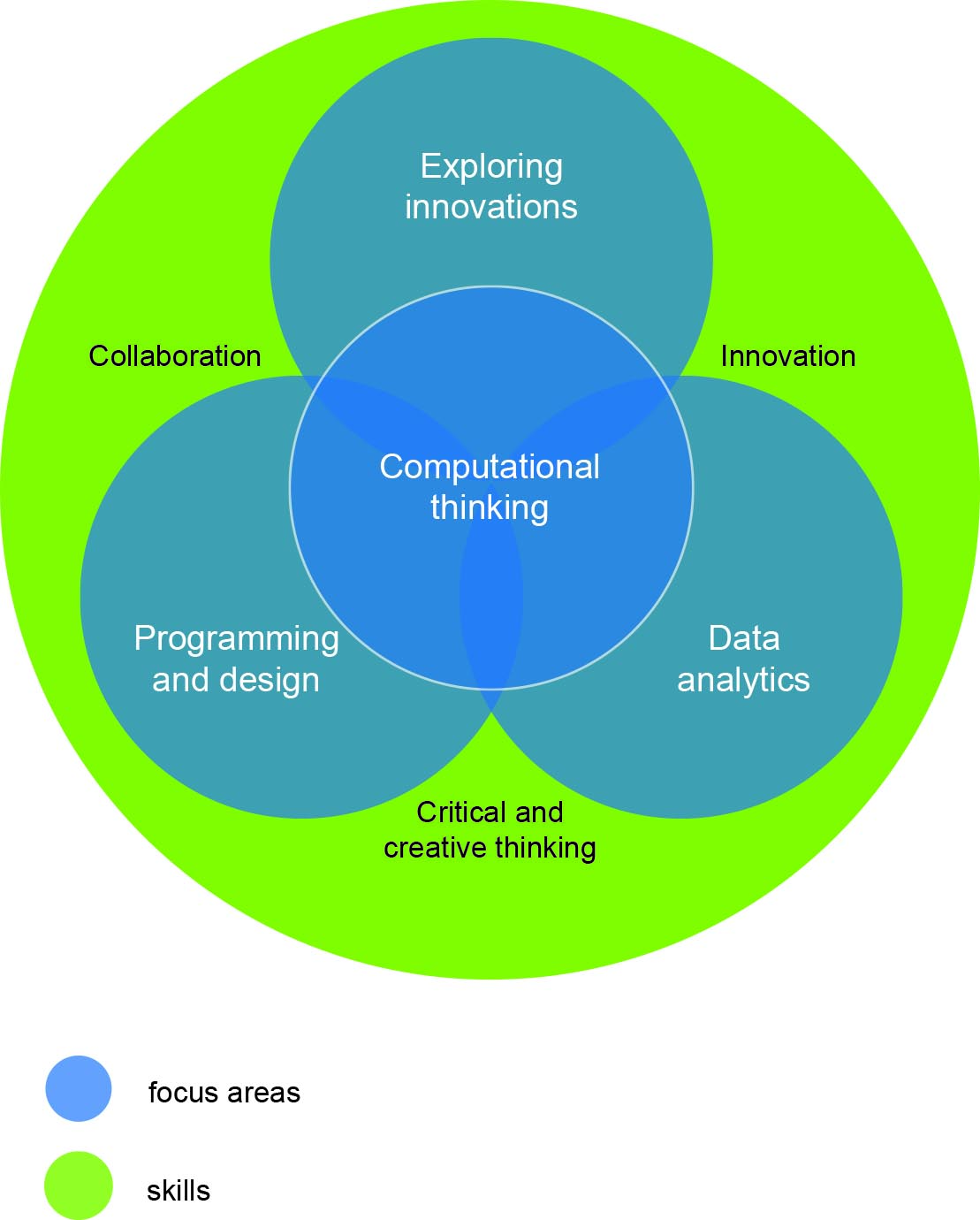
Digital Technologies provides a range of contexts for students to extend their creative dispositions and skills in exploration, generation, design, and creation. They generate and explore problems of interest and actively seek answers to questions posed by the problem. In extending their creative thinking, students challenge assumptions and critically evaluate information relating to problems of interest. They imagine possibilities and make connections in their learning to generate, develop, and refine ideas, and to determine solutions. With initiative, imagination, and perseverance, students scope, design, and create innovative solutions to problems of interest.

Students work collaboratively to problem-solve and create innovative digital solutions to problems of interest. Collaboration enables students to develop their social and cognitive skills and ways of combining the knowledge and skills of group members to solve problems effectively as a team. Students extend and apply their personal and social capability as they actively engage with and respect the different perspectives of others. Students work together in interconnected ways as they move between activities and contribute to and complete activities begun by others. Through collaboration students make connections in their critical and creative thinking, decision-making, and problem-solving. Collaboration may include the use of synchronous and/or asynchronous digital communication strategies.

Innovation in Digital Technologies includes students generating and articulating original ideas, creating new processes, products, and solutions, and designing solutions that improve existing ideas, processes, and products. Students extend the range and combination of skills that enable them to contribute to innovation in digital solutions. These skills encompass academic, technical, and soft skills, and the ability to apply these skills and knowledge to solving unfamiliar problems. Students work independently and/or collaboratively to generate ideas and create innovative solutions and creative products. They extend their skills in critical and creative thinking and problem-solving, and make connections in their learning across disciplines to generate ideas and create innovative digital solutions. In the pursuit of innovation students increase their willingness to take risks and appreciate the value of learning from what does not work, as well as from what does work, as they scope and design innovative solutions.

The following is a diagrammatical representation of the integration of the focus areas and skills.

Digital Technologies



DRAFT — for teaching in 2019

Focus area 1: Computational thinking

Students develop computational thinking skills and strategies to identify, deconstruct, and solve problems of interest. These strategies include decomposition, pattern recognition, abstraction, and algorithm design.

Students identify and deconstruct problems, including complex problems, into logical sets of sequential or iterative parts using tools that show dependencies and flow of sequence between parts. Complex problems are questions or issues that cannot be answered through simple logical procedures alone. They generally also require abstraction to deconstruct the problem and focus on core concepts and ideas.

Students develop skills in identifying patterns of similarities or repetition in data or in a series of possible solutions, which can then be applied in algorithm design, including creating generalised algorithms.

Abstraction involves removing irrelevant detail to leave only the key elements of a problem, thereby simplifying the problem. Through abstraction students identify common elements and make connections with other simplified problems.

Students design algorithms to produce an output. An algorithm might be written in pseudocode, or represented in a structure chart or flow chart.

To support their understanding of designing solutions for problems, students use computational thinking skills to analyse and evaluate relevant data, test hypotheses, make evidence-based decisions, and revisit steps using new information to update their analyses. A solution may take the form of a product, prototype, and/or proof of concept.

Self-assessment tools or skills frameworks may be used to support the development and application of students’ skills in working collaboratively.

The following framework provides a set of possible techniques and strategies that can be used for learning.

| Key learning elaboration | Possible techniques and strategies |
| --- | --- |
| **Decomposition** |  |
| Students identify and deconstruct problems, including complex problems, into logical sets of sequential or iterative parts using tools that show dependencies and flow of sequence between parts. | Students identify and deconstruct a problem logically into two or more sequential tasks using tools that show dependencies and flow of sequence between parts.  Students apply logical thinking to deduce as much information as possible from the initial information provided, and revisit steps based on new information. |
| **Pattern recognition** |  |
| Students identify patterns of similarities or repetition in data or in a series of possible solutions, which can then be applied in algorithm design. | Students find repeating patterns in data and/or an algorithmic solution to simplify through iteration and implementation reuse.  Students recognise that there are problems that will develop into similar algorithms, and that the same patterns can apply in other areas. |
| **Abstraction** |  |
| Students use recognised patterns to identify general rules and concepts. | Students identify common elements and connections with other problems.  Students use levels of abstraction to create generalised rules and concepts.  Students map different scenarios to the same rules and concepts. |
| **Algorithm design** |  |
| Students design ordered instructions to produce an output. Such instructions are called ‘algorithms’. | Students experiment with core algorithms such as searching and sorting.  Students run an initial, informal test of the algorithm to understand and verify it.  Students graphically represent algorithms at a high level of detail, for example, by creating structure charts or flow charts. |

| Key learning elaboration | Possible techniques and strategies |
| --- | --- |
| **Coding skills** |  |
| Students create algorithms, which may be written in pseudocode, or represented in a structure chart or flow chart. | Students use pseudocode to help develop algorithms that allow a task to be performed.  Students apply basic constructs that control algorithmic flow including sequence, selection, and iteration.  Students apply pattern recognition and abstraction to develop algorithms to perform specific tasks.  Students design sets of algorithms to build simple prescribed applications.  Students check an algorithm for accuracy and amend its design in response to failed test cases. |

Focus area 2: Design and programming

Students work individually and collaboratively to create practical, innovative solutions to problems of interest. They use computational thinking strategies to develop a series of related instructions that can be executed by computer hardware. Students design and check a solution, and ensure that all design considerations have been met. They code the program using a general-purpose programming language (GPL) and validate the solution. A solution may take the form of a product, prototype, and/or proof of concept.

Students develop and apply their understanding of program design, pseudocode, simple constructs, array processing, and modularisation. Some students may also wish to explore object-oriented design.

Students review, develop, and extend the building blocks of a GPL (variables, expressions, assignment, and input/output commands), and the simple constructs of sequence, selection, and iteration.

Students describe the purpose of the code using comments (annotations) so that another person can read, understand, apply, and/or edit the program or algorithm.

Students practise testing techniques by fixing syntactic and runtime errors, and they correct logic errors by creating and executing test cases. Students explore software development tools to support code development and maintenance.

Self-assessment tools or skills frameworks may be used to support the development and application of students’ skills in working collaboratively.

The following framework provides a set of possible techniques and strategies that can be used for learning.

| Key learning elaboration | Possible techniques and strategies |
| --- | --- |
| **Design thinking** |  |
| Students design a solution. | Students use computational thinking skills to design a solution prior to coding. |
| **Programming skills** |  |
| Students are introduced to software-development tools to support code development. | Students use software-development tools such as integrated development environments (IDEs), debuggers, and compilers/interpreters to support code development |
| Students code the program using a general-purpose programming language (GPL). | Students develop code feature by feature.  Students review, develop, and extend the building blocks of a GPL (variables, expressions, assignment, and input/output commands), and the simple constructs of sequence, selection, and iteration.  Students review and use basic data structures (strings, numbers, Booleans, and arrays).  Students describe the purpose of the code using comments (annotations) so that another person can read, understand, apply, and/or edit the program or algorithm. |
| **Testing and maintenance** |  |
| Students practise testing techniques by fixing syntactic and runtime errors, and logic errors, by creating and executing test cases. | Students test code feature by feature.  Students practise testing and debugging methods and skills: fix syntactic/runtime errors and develop and execute test cases to identify logic errors. |
| Students validate the solution. | Students develop code using recommended practices and conventions (e.g. indenting, naming, commenting, using constants, modularisation). |
| Students explore software-development tools to support code development and maintenance. | Students apply version control and file management techniques. |

Focus area 3: Data analytics

Students use a range of data-collection tools, techniques, and methods to collect and/or source data. They determine the appropriate data collection method for a project.

Students apply computational thinking skills to collect, extract, interpret, and model data sets. They analyse data sets to identify social, economic, environmental, scientific, and/or other trends. These trends provide a source of information from which students can explore and test hypotheses to determine solutions to issues, for example, in business, industry, the environment, and the community. Such issues may be of local, national, and/or global importance.

Students analyse data sets to find simple relationships, which are generally linear. They may also analyse more complex relationships, which may be, for example, exponential or logarithmic.

Students study the ethical implications of data collection, storage, sharing, use, and security. They examine privacy issues and the role of government, business, and society in determining cybersecurity policy. Students extend their critical understanding of the appropriate and ethical uses of digital technology.

Self-assessment tools or skills frameworks may be used to support the development and application of students’ skills in working collaboratively.

The following framework provides a set of possible techniques and strategies that can be used for learning.

| Key learning elaboration | Possible techniques and strategies |
| --- | --- |
| **Sources of data** |  |
| Students identify trends, for example, social, economic, environmental, scientific, and other trends, through extracting, interpreting, and modelling data sets. | Students explore a research question or hypothesis.  Students design and conduct surveys.  Students extract data from online and public data sets.  Students make use of digital technologies to collect data; for example, video cameras, wearable technology, sensors (e.g. light, body heat, and movement), GPS tracking, and dynamic and static screen-capture.  Students check data integrity including: currency, authenticity, relevance, accuracy, and outliers (cleaning). |
| **Analysis** |  |
| Students analyse data sets to inform thinking about issues related to trends. | Students investigate characteristics of a data set, such as shape, through visualisation, using a range of tools. |
| Students analyse data sets to find simple relationships, which are generally linear. They may also analyse more complex relationships, which may be, for example, exponential or logarithmic. | Students test a research question or hypothesis.  Students recognise patterns and relationships in data.  Students use simple quantitative techniques to analyse data and tabulate results.  Students explore examples of how data analytics are used, for example, in business, health, science, and government to support decision-making. |
| **Ethical considerations** |  |
| Students evaluate the ethical implications surrounding the collection, storage, use, and/or security of data. | Students evaluate strategies for use by government, business, organisations, and individuals for:   * collection of personal data * data security and storage * data protection and backup * privacy and anonymising data * cyber security. |

Focus area 4: Iterative project development

Students identify a concept or problem of interest and engage collaboratively and individually in iterative project development, where a product is developed in stages by applying techniques such as agile practices and/or iterative engineering design processes.

They scope, create, test, and evaluate a proposed digital solution to a problem of interest. The solution may be a product, prototype or proof of concept and should include innovative features.

When working collaboratively, students choose a problem that is of interest to the group and relevant to the school and/or the local community. The problem chosen by the group should be manageable. Students identify the project client or clients (stakeholders), and engage with them at key stages of the project.

Students deconstruct the problem, scope a preliminary design for the digital solution, and identify roles and activities for group members. They develop and agree on a plan based on time-scale and resources, communication strategies to be used, and key features of the project deliverables. Students have the opportunity to move between activities and to contribute and complete activities begun by others.

Students work collaboratively to collect and analyse data relevant to the problem; drawing conclusions and/or making predictions that inform the process of scoping and designing the solution. They identify one or more project clients from either the school or the community, and communicate with the client or clients at key stages of the project.

Students design and fully develop each feature of the software product or prototype, using design and programming skills. Features are deliverables and students develop the highest priority features first. The activities and actions of each student are connected and build on those of other group members. Students are responsive to contributions made by others, recognising that engaging with different perspectives enhances their own knowledge, understanding, and solutions.

At each iteration, through cooperation and use of digital and other communication strategies, students clarify, amend, and/or add one or more features based on their learning from the previous iterations. They may use a range of forms to document and clarify amendments and additions to requirements, such as storyboards, screenshots, visuals, sketches, wireframe models, or user stories. Students test each feature to ensure that it is both usable and ready to progress to the next iteration.

Students research and discuss the ethical implications of their digital solution for individuals, groups, societies, and/or the environment.

Self-assessment tools or skills frameworks may be used to support the development and application of students’ skills in working collaboratively.

The following framework provides a set of possible techniques and strategies that can be used for learning.

| Key learning elaboration | Possible techniques and strategies |
| --- | --- |
| **Problem scoping and planning** |  |
| Students engage in iterative project development, where a product is developed in stages by applying techniques such as agile practices and/or iterative engineering design. | Students form project teams.  Students conduct ‘market research’ to determine existing solutions to a problem. |
| Students add to or amend one or more features and/or requirements of the project as the project evolves and conditions change. | Students define an original concept or problem by identifying one or more relevant data sets.  Students identify audience and purpose.  Students create a set of requirements that identify key features of a digital solution, with opportunities to innovate.  Students clarify requirements, based on input from a key stakeholder/user representative.  Students prioritise feature development.  Students define an agreed plan of team activities and a project time-scale based on time available, communication strategies, and key features of deliverables.  Students identify, negotiate, and allocate roles in the project team and activities to be undertaken by each team member.  Students agree on and apply digital and other communication strategies to enable participation and contributions from group members.  Students select resources such as software, hardware, and media.  Students identify strategies to minimise potential risks to things such as the time-scale for deliverables.  Students consider how solutions that are created now will be used in the future. |
| **Design thinking** |  |
| Students learn and apply project-design skills. | Students turn an idea into a solution such as a dynamic website, application program, wearable technology, or other digital solution. |
| Students develop the highest priority features first. | Students monitor progress against time-scale, risks, and resources. |
| Students produce a usable product at the end of each iteration. | Students maintain effective communication within the group and between the group and key stakeholders.  Students make improvements based on user testing/feedback.  Students modify agreed plan to reflect improvements.  Students evaluate whether the deliverable is consistent with the specified requirements, and make further improvements as needed. |
| **Code development and evaluation** |  |
| Students fully develop and test each feature before moving to the next feature. | Students develop code for each feature and integrate it with the existing design (version control).  Students evaluate feature by feature with system testing and user testing. |
| **Ethical considerations** |  |
| Students research and discuss the ethical implications of their digital solution for individuals, groups, societies, and/or the environment. | Students consider accessibility factors, such as visibility, access of use, device access, cost, and language.  Students consider security, encryption, and privacy issues (e.g. personal data and images). |

Assessment scope and requirements

All Stage 2 subjects have a school assessment component and an external assessment component.

Evidence of learning

The following assessment types enable students to demonstrate their learning in Stage 2 Digital Technologies.

School assessment (70%)

* Assessment Type 1: Project Skills (50%)
* Assessment Type 2: Collaborative Project (20%)

External assessment (30%)

* Assessment Type 3: Individual Digital Solution (30%).

Students should provide evidence of their learning through six assessments, including the external assessment component. Students undertake:

* four project skills tasks
* one collaborative project
* one individual digital solution.

Assessment design criteria

The assessment design criteria are based on the learning requirements and are used by:

* teachers to clarify for students what they need to learn
* teachers and assessors to design opportunities for students to provide evidence of their learning at the highest possible level of achievement.

The assessment design criteria consist of specific features that:

* students should demonstrate in their learning
* teachers and assessors look for as evidence that students have met the learning requirements.

For this subject the assessment design criteria are:

* computational thinking
* development and evaluation
* research and ethics.

The specific features of these criteria are described below.

The set of assessments, as a whole, must give students opportunities to demonstrate each of the specific features by the completion of study of the subject.

Computational Thinking

The specific features are as follows:

CT1 Application of computational thinking concepts and techniques to identify and deconstruct problems of interest.

CT2 Use of abstraction to identify core concepts and ideas.

CT3 Analysis of relationships in data sets to draw conclusions and make predictions.

CT4 Application of skills and processes to develop solutions to problems of interest.

Development and Evaluation

The specific features are as follows:

DE1 Design and creation of digital solutions or a prototype.

DE2 Application of iterative development, testing, modification, and documentation of a digital solution or prototype.

DE3 Evaluation of the effectiveness of a digital solution or prototype.

DE4 Explanation, with supporting evidence, of own role in and contribution to projects.

Research and Ethics

The specific feature is as follows:

RE1 Research into and discussion of the ethical considerations in digital technologies.

School assessment

Assessment Type 1: Project Skills (50%)

Students produce four project skills tasks in which they examine approaches to identifying, deconstructing, and solving problems of interest by applying:

* computational thinking skills, including abstraction
* data analysis skills
* design and programming skills
* iterative project-development techniques.

The problems chosen should be of interest to the students.

At least one of the tasks should involve students working collaboratively, with each student providing individual evidence of their role in and contribution to the collaborative task.

At least one of the tasks should focus on students analysing simple and complex data sets related to a problem of interest to identify patterns and/or trends, draw conclusions, and make predictions.

At least one of the tasks should involve an assessment of programming skills.

In at least one of the tasks, students research and discuss the ethical implications of data use and/or digital solutions for individuals, groups, societies, and/or the environment.

The tasks should be presented in multimodal form. Together, the four tasks should be the equivalent in multimodal form of a maximum of 20 minutes.

Tasks may include, for example:

* a screen-capture validation, highlighting innovative features
* a multimedia presentation of a solution to a problem, analysis of data, or prediction of trends
* an annotated prototype solution or proof of concept
* an informed debate about the ethics of data storage and use.

For this assessment type, students provide evidence of their learning primarily in relation to the following assessment design criteria:

* computational thinking
* development and evaluation
* research and ethics.

Assessment Type 2: Collaborative Project (20%)

Students apply their learning about iterative project development to create a digital solution through a collaborative project. Each student presents individual evidence of their contribution to the project.

Students create:

* a digital solution (product, prototype, or proof of concept)
* an explanation of the solution
* an individual evaluation of each student’s role in developing the solution, and of the effectiveness of the collaborative working process.

The digital solution should be no more than 1 GB.

The problem chosen should of interest to the group, the school, and/or the local community.

Students, as a group, present the project to the client or clients. They explain the digital solution (product, prototype, or proof of concept) and their project evaluation. Each student in the group explains their role in and contribution to the project, supported by evidence. Evidence may include digital communications between group members, screenshots, and/or any other relevant information. The explanation and evaluation should be up to 5 minutes per student.

The digital solution should be presented in a digital or multimodal form. The explanation and evaluation should be oral and multimodal and be recorded.

The teacher assesses each student’s individual contribution to the digital solution, explanation, and evaluation.

For this assessment type, students provide evidence of their learning primarily in relation to the following assessment design criteria:

* computational thinking
* development and evaluation.

External assessment

Assessment Type 3: Individual Digital Solution (30%)

Students apply iterative project techniques to independently identify, deconstruct, and solve a problem of interest by creating and evaluating a digital solution or prototype.

The problem should be chosen by, and be of interest to, the student. The problem should be manageable and have sufficient complexity to enable the student to achieve at the highest level. The problem may take as a starting point an aspect of a problem or solution created for Assessment Type 1 or Assessment Type 2, but must not repeat work already submitted for assessment.

In creating their digital solution or prototype, students should be mindful of any ethical considerations.

The solution or prototype should include:

* original source code and/or adapted code displaying selection, repetition, and sequencing, accompanied by design comments
* algorithm design
* graphical user interface and/or instructions for use.

The solution or prototype must be supported by a designer’s statement that discusses:

* the effectiveness of the solution or prototype
* a feature or features that could be considered innovative in solving the problem.

The individual digital solution and supporting documentation (code, design comments, graphical user interface and/or instructions for use) should be submitted in multimodal form.

The digital solution or prototype should be no more than 1 GB.

The designer’s statement should be a maximum of 3 minutes if oral, 500 words if written, or the equivalent if multimodal.

The following specific features of the assessment design criteria for this subject are assessed in Assessment Type 3: Individual Digital Solution:

* computational thinking — CT1, CT2, CT4
* development and evaluation — DE1, DE2, DE3

Performance standards

The performance standards describe five levels of achievement, A to E.

Each level of achievement describes the knowledge, skills, and understanding that teachers and assessors refer to in deciding how well students have demonstrated their learning on the basis of the evidence provided.

During the teaching and learning program the teacher gives students feedback on their learning, with reference to the performance standards.

The student’s results for school assessment and external assessment are combined for a final result, which is reported as a grade between A and E.

Performance Standards for Stage 2 Digital Technologies

| - | Computational Thinking | Development and Evaluation | Research and Ethics |
| --- | --- | --- | --- |
| A | Astute and creative application of computational thinking concepts and techniques to clearly identify and deconstruct problems of interest.  Insightful use of abstraction to identify core concepts and ideas.  In-depth analysis of relationships in data sets to draw insightful conclusions and make well-justified predictions.  Highly purposeful application of skills and processes to develop highly efficient and logical solutions to complex problems of interest. | Clear and consistent use of initiative in the design and creation of digital solution or prototype that includes innovative features.  Highly purposeful and strategic application of iterative development, testing, modification, and documentation of an innovative digital solution or prototype.  Insightful evaluation of the effectiveness of a digital solution or prototype.  Insightful explanation, supported by clear and highly convincing evidence of own role in and contribution to projects. | In-depth research and discussion of the ethical considerations in digital technologies. |
| B | Well-considered application of computational thinking concepts and techniques to identify and deconstruct problems of interest.  Some insights in the use of abstraction to identify core concepts and ideas.  Some depth in analysis of relationships in data sets to draw informed conclusions and make justified predictions.  Purposeful application of skills and processes to develop efficient and mostly logical solutions to some complex problems of interest. | Mostly consistent use of initiative in the design and creation of digital solution or prototype that includes one or more innovative features.  Mostly purposeful application of iterative development, testing, modification, and documentation of a digital solution or prototype, with some innovation.  Well-considered evaluation of the effectiveness of a digital solution or prototype.  Some depth in explanation, supported by clear and mostly convincing evidence of own role in and contribution to projects. | Some depth in research and discussion of the ethical considerations in digital technologies. |
| C | Application of computational thinking concepts and techniques to identify and deconstruct problems of interest.  Some use of abstraction to identify core concepts and ideas.  Description, with some analysis, of relationships in data sets to draw generally informed conclusions and make predictions, with some justification.  Application of skills and processes to develop generally efficient and logical solutions to problems of interest. | Some use of initiative in the design and creation of digital solution or prototype, which may include one or more innovative features.  Competent application of iterative development, testing, modification, and documentation of a digital solution or prototype, with one or more innovative features.  Description of the effectiveness of a digital solution or prototype, with evaluation of some features.  Explanation, supported by generally clear evidence, of own role in and contribution to projects. | Considered research and discussion of the ethical considerations in digital technologies. |
| D | Partial application of basic computational thinking concepts and techniques to identify and describe problems of interest.  Identification and description of some basic core concepts and/or ideas.  Identification and use of one or more simple relationships in data sets to draw a partial conclusion and/or make a prediction based on limited evidence.  Partial application of skills and processes to develop solutions to simple problems of interest. | Partial design and creation of digital solution or prototype.  Basic application of some iterative development, testing, modification, and/or documentation of a digital solution or prototype.  Partial description of the effectiveness of a digital solution or prototype.  Basic explanation of own role in and/or contribution to projects, with limited supporting evidence. | Basic research and discussion of one or more ethical considerations in digital technologies. |
| E | Attempted application of a limited number of basic computational thinking concepts or techniques to describe a problem of interest.  Attempted identification and description of a core concept or idea.  Attempted use of limited, simple data sets to draw a conclusion or make a prediction.  Attempted application of skills and processes to develop partial solutions to some simple problems of interest. | Attempted design and creation of digital solution or prototype.  Attempted application of simple iterative development, testing, modification, or documentation of a digital solution or prototype.  Limited description of a digital solution or prototype.  Limited description of own participation in projects. | Attempted research and discussion of ethical considerations in digital technologies. |

Assessment integrity

The SACE Assuring Assessment Integrity Policy outlines the principles and processes that teachers and assessors follow to assure the integrity of student assessments. This policy is available on the SACE website (www.sace.sa.edu.au) as part of the SACE Policy Framework.

The SACE Board uses a range of quality assurance processes so that the grades awarded for student achievement, in both the school assessment and the external assessment, are applied consistently and fairly against the performance standards   
for a subject, and are comparable across all schools.

Information and guidelines on quality assurance in assessment at Stage 2 are available on the SACE website (www.sace.sa.edu.au).

Support materials

Subject-specific advice

Online support materials are provided for each subject and updated regularly on the SACE website (www.sace.sa.edu.au). Examples of support materials are sample learning and assessment plans, annotated assessment tasks, annotated student responses, and recommended resource materials.

Advice on ethical study and research

Advice for students and teachers on ethical study and research practices is available in the guidelines on the ethical conduct of research in the SACE, which are on the SACE website (www.sace.sa.edu.au).