

Accreditation  
Period  
Units 1–4  
**2025–2028**

Victorian Certificate of Education

# Applied Computing

Study Design

## Amendments to study design history

Version	Status	Release Date	Comments
1.1	Current	May 2024	Update to second bullet point for Unit 1, Outcome 1 (page 29).
1.0	Superseded	March 2024	Original study design.

Authorised and published by the Victorian Curriculum and Assessment Authority  
Level 7, 2 Lonsdale Street  
Melbourne VIC 3000

ISBN: 978-1-923025-75-2

© Victorian Curriculum and Assessment Authority 2024

No part of this publication may be reproduced except as specified under the *Copyright Act 1968* or by permission from the VCAA. Excepting third-party elements, schools may use this resource in accordance with the [VCAA educational allowance](#). For more information read the [VCAA copyright policy](#).

The VCAA provides the only official, up-to-date versions of VCAA publications. Details of updates can be found on the [VCAA website](#).

This publication may contain copyright material belonging to a third party. Every effort has been made to contact all copyright owners. If you believe that material in this publication is an infringement of your copyright, please email the [Copyright Officer](#).

Copyright in materials appearing at any sites linked to this document rests with the copyright owner(s) of those materials, subject to the Copyright Act. The VCAA recommends you refer to copyright statements at linked sites before using such materials.

VCE and the VCAA logo are registered trademarks of the Victorian Curriculum and Assessment Authority.

Contact us if you need this information in an accessible format: for example, large print or audio.

Telephone (03) 9032 1635 or email [vcaa.media.publications@education.vic.gov.au](mailto:vcaa.media.publications@education.vic.gov.au)

# Contents

<b>Important information .....</b>	<b>5</b>
<b>Introduction.....</b>	<b>6</b>
Scope of study.....	6
Rationale .....	6
Aims.....	7
Structure .....	7
Entry .....	7
Duration .....	7
Changes to the study design.....	7
Monitoring for quality .....	8
Safety and wellbeing .....	8
Requirements for teaching standard .....	8
Use of digital resources .....	8
Employability skills .....	8
Legislative compliance .....	9
Child Safe Standards .....	9
<b>Assessment and reporting .....</b>	<b>10</b>
Satisfactory completion .....	10
Levels of achievement.....	10
Authentication.....	11
Terms used in this study .....	12
<b>Units 1 to 4: Problem-solving methodology .....</b>	<b>17</b>
<b>Units 1 to 4: Problem-solving methodology specifications.....</b>	<b>18</b>
<b>Unit 1: Applied computing.....</b>	<b>24</b>
Area of Study 1.....	24
Area of Study 2.....	27
Assessment .....	28
<b>Unit 2: Applied computing.....</b>	<b>30</b>
Area of Study 1.....	30
Area of Study 2.....	33
Assessment .....	36
<b>Unit 3: Data analytics .....</b>	<b>37</b>

Area of Study 1 .....	37
Area of Study 2.....	40
School-based assessment .....	42
School-assessed Task .....	43
External assessment .....	43
<b>Unit 4: Data analytics .....</b>	<b>44</b>
Area of Study 1.....	44
Area of Study 2.....	46
School-based assessment .....	49
School-assessed Task .....	51
External assessment .....	52
<b>Unit 3: Software development .....</b>	<b>53</b>
Area of Study 1.....	53
Area of Study 2.....	56
School-based assessment .....	59
School-assessed Task .....	60
External assessment .....	60
<b>Unit 4: Software development .....</b>	<b>61</b>
Area of Study 1.....	61
Area of Study 2.....	63
School-based assessment .....	65
School-assessed Task .....	67
External assessment .....	68

# Important information

## Accreditation period

Units 1–4: 1 January 2025 – 31 December 2028

Implementation of this study commences in 2025.

## Other sources of information

The [VCAA Bulletin](#) is the only official source of changes to regulations and accredited studies. The *Bulletin* also regularly includes advice on VCE studies. It is the responsibility of each VCE teacher to refer to each issue of the *Bulletin*. The *Bulletin* is available as an e-newsletter via [free subscription](#) on the VCAA website.

To assist teachers in developing courses, the VCAA publishes online [Support materials](#) (incorporating the content previously supplied in the *Advice for teachers*).

The current [VCE Administrative Handbook](#) contains essential information on assessment processes and other procedures.

## VCE providers

Throughout this study design the term 'school' is intended to include both schools and other VCE providers.

## Copyright

Schools may use this VCE study design in accordance with the VCAA educational allowance, including making copies for students and use within the school's secure online teaching and learning environment. For details, access the full [VCAA copyright policy](#) on the VCAA website.

# Introduction

## Scope of study

Individuals, organisations and governments are adopting new and emerging technologies faster than ever before. As a result, the demand for skilled professionals that can support the adoption of these technologies is also increasing. VCE Applied Computing focuses on four interrelated disciplines where skills are in high demand and will continue to be for the foreseeable future. These disciplines are data analytics, programming, emerging and innovative technologies, and cyber security.

Pivotal to this are the strategies and techniques for creating digital solutions to meet identified problems, needs and opportunities. VCE Applied Computing focuses on students learning to apply a range of thinking skills within local, national and global contexts. The study incorporates computational, design and systems thinking, as part of students creating digital solutions. Legal and ethical responsibilities are also considered at an individual and organisational level to ensure solutions protect intellectual property and the privacy of data and information.

Through a structured approach to problem-solving, incorporating the disciplines and critical and creative thinking strategies, students develop a robust awareness of the technical, social and economic impacts of the digital solutions being developed and used around the world, now and in the future.

## Rationale

As technology continues to evolve rapidly, there are increasing opportunities for enterprising and entrepreneurial individuals to develop new technologies, as well as identify innovative uses for existing technologies. Students develop a range of data analytics, programming and cyber security skills that are in high demand in today's digital age.

VCE Applied Computing equips students with the knowledge and skills required to navigate and adapt within a dynamic technological landscape, use emerging technologies, envisage new uses for digital tools and consider the benefits to society at a local, national and global level.

VCE Applied Computing facilitates student-centred learning that enables students to build capabilities in their critical and creative thinking, communicate and collaborate with their peers, and develop personal, social and digital literacy skills. Students are provided with practical opportunities and choices to create digital solutions for real-world problems in a range of settings.

VCE Applied Computing plays a crucial role in transforming and improving student outcomes. It provides students with diverse pathways that enable them to develop into knowledgeable and skilled individuals who have the capability to work collaboratively in this rapidly expanding technological landscape. VCE Applied Computing leads directly to further studies in areas such as artificial intelligence, computer science, cyber security, data analytics and data science, data management, digital literacy, games development, networks, programming, robotics, software engineering, systems analysis and telecommunications. The skills acquired in VCE Applied Computing could be utilised across a range of industries including healthcare, finance, entertainment, education, construction and business/commerce.

## Aims

This study enables students to:

- understand how digital systems and solutions can be used by individuals and organisations
- develop an understanding of the roles and applications of data analytics, programming, cyber security and emerging technologies within society
- build a capacity to solve problems by applying the problem-solving methodology to analyse problems, needs and opportunities, design and develop solutions and evaluate how effectively solutions meet needs
- apply project management techniques to assist with the development of digital solutions
- develop an informed perspective on current and emerging digital tools and trends
- identify and evaluate innovative and emerging opportunities for digital technologies
- develop critical and creative thinking, communication and collaboration, and personal, social and digital literacy skills.

## Structure

The study is made up of six units.

- Unit 1: Applied computing (IT011)
- Unit 2: Applied computing (IT012)
- Unit 3: Data analytics (IT023)
- Unit 4: Data analytics (IT024)
- Unit 3: Software development (IT033)
- Unit 4: Software development (IT034)

Note: students may elect to undertake one or both Unit 3 and 4 sequences.

Each unit deals with specific content contained in areas of study and is designed to enable students to achieve a set of outcomes for that unit. Each outcome is described in terms of key knowledge and key skills.

## Entry

There are no prerequisites for entry to Units 1, 2 and 3. Students must undertake Unit 3 and Unit 4 as a sequence. Units 1–4 are designed to the equivalent standard of the final two years of secondary education. All VCE studies are benchmarked against comparable national and international curriculums.

## Duration

Each unit involves at least 50 hours of scheduled classroom instruction.

## Changes to the study design

During its period of accreditation, minor changes to the study will be announced in the [VCAA Bulletin](#). The *Bulletin* is the only source of changes to regulations and accredited studies. It is the responsibility of each VCE teacher to monitor changes or advice about VCE studies published in the *Bulletin*.

## Monitoring for quality

As part of ongoing monitoring and quality assurance, the VCAA will periodically undertake an audit of VCE Applied Computing to ensure the study is being taught and assessed as accredited. The details of the audit procedures and requirements are published annually in the [VCE Administrative Handbook](#). Schools will be notified if they are required to submit material to be audited.

## Safety and wellbeing

It is the responsibility of the school to ensure that duty of care is exercised in relation to the health and safety of all students undertaking the study.

## Requirements for teaching standard

The Principal must make sure that students have access to adequate facilities and resources to complete any VCE study they are offered. To ensure students are able demonstrate the outcomes of this study, students should have access to digital resources as outlined in this study design. There are no requirements for specialist facilities and [teacher qualifications](#) specified for this study.

## Use of digital resources

Students require access to the following resources to be able to demonstrate the outcomes:

- appropriate laptop, notebook or desktop computers
- printer(s)
- internet access
- a range of software or web-based applications including:
  - relational database management software (RDBMS) to store data in tables and perform queries
  - spreadsheet software tools to store data and perform a range of calculations, including statistical calculations
  - software tools that can run Structured Query Language (SQL) queries to extract data
  - software tools to create a range of data visualisations
  - software tools to create multimodal presentations
  - drawing and graphic software tools to create diagrams
  - programming tools and/or integrated development environments to facilitate programming and testing of modules and solutions
  - software tools capable of documenting and monitoring a project plan
  - software tools capable of documenting ideation and designs
  - software tools for designing solutions.

## Employability skills

This study offers a number of opportunities for students to develop employability skills. The [Support materials](#) provide specific examples of how students can develop employability skills during learning activities and assessment tasks.



## Legislative compliance

When collecting and using information, the provisions of privacy and copyright legislation, such as the Victorian *Privacy and Data Protection Act 2014* and *Health Records Act 2001*, and the federal *Privacy Act 1988* (Cwlth) and *Copyright Act 1968* (Cwlth), must be met.

## Child Safe Standards

Schools and education and training providers are required to comply with the Child Safe Standards made under the Victorian *Child Wellbeing and Safety Act 2005*. Registered schools are required to comply with *Ministerial Order No. 1359 Implementing the Child Safe Standards – Managing the Risk of Child Abuse in Schools and School Boarding Premises*. For further information, consult the websites of the [Victorian Registration and Qualifications Authority](#), the [Commission for Children and Young People](#) and the [Department of Education](#).

# Assessment and reporting

## Satisfactory completion

The award of satisfactory completion for a unit is based on the teacher's decision that the student has demonstrated achievement of the set of outcomes specified for the unit. Demonstration of achievement of outcomes and satisfactory completion of a unit are determined by evidence gained through the assessment of a range of learning activities and tasks.

Teachers must develop courses that provide appropriate opportunities for students to demonstrate satisfactory achievement of outcomes.

The decision about satisfactory completion of a unit is distinct from the assessment of levels of achievement. Schools will report a student's result for each unit to the VCAA as S (satisfactory) or N (not satisfactory).

## Levels of achievement

### Units 1 and 2

Procedures for the assessment of levels of achievement in Units 1 and 2 are a matter for school decision. Assessment of levels of achievement for these units will not be reported to the VCAA. Schools may choose to report levels of achievement using grades, descriptive statements or other indicators.

### Units 3 and 4

The VCAA specifies the assessment procedures for students undertaking scored assessment in Units 3 and 4. Designated assessment tasks are provided in the details for each unit in VCE study designs.

The student's level of achievement in Units 3 and 4 will be determined by School-assessed Coursework (SAC), a School-assessed Task (SAT) as specified in the VCE study design, and external assessment.

The VCAA will report the student's level of achievement on each assessment component as a grade from A+ to E or UG (ungraded). To receive a study score, the student must achieve two or more graded assessments in the study and receive an S for both Units 3 and 4. The study score is reported on a scale of 0–50; it is a measure of how well the student performed in relation to all others who completed the study. Teachers should refer to the current [VCE Administrative Handbook](#) for details on graded assessment and calculation of the study score.

Percentage contributions to the study score in VCE Applied Computing are as follows:

- Units 3 and 4 School-assessed Coursework: 20 per cent
- Units 3 and 4 School-assessed Task: 30 per cent
- end-of-year examination: 50 per cent.

Details of the assessment program are described in the sections on Units 3 and 4 in this study design.

## Authentication

Work related to the outcomes of each unit will be accepted only if the teacher can attest that, to the best of their knowledge, all unacknowledged work is the student's own. Teachers need to refer to the current [VCE Administrative Handbook](#) for authentication rules and strategies.

## Terms used in this study

For the purposes of this study design and associated assessment, the following definitions will apply.

Term	Definition
Alpha testing	A testing phase that checks whether modules or solutions meet all requirements and function as expected. Alpha testing is carried out by developers, independent testers or high-level users in a development or testing environment throughout the development phase.
Archiving	The process of moving data from a system that no longer needs to be accessed regularly to a less frequently accessed storage area for future use or to meet compliance requirements. This ensures that data can be stored separately to systems for long periods of time without impacting on current performance or storage requirements.
Backup	The process of making a copy of data and storing the copy separately to the original data in case it is needed due to data loss. Backed up data can be full (entire copy of data), differential (changes since last full backup), incremental (changes since last backup) or a combination of these. Backups can either run manually or be scheduled to run automatically, and can be stored on a local hard drive (distinct from the original source), on external storage devices or by using cloud computing. Backups are restored when data loss occurs.
Beta testing	A testing phase that checks whether solutions meet all requirements, function as expected, are stable and are usable by intended user groups. Beta testing is performed in a development, testing or production environment with hardware identical (or similar) to that on which it will be implemented on by users of the solution. Feedback from beta testing can be used to recommend or make modifications to modules or solutions.
Computational thinking	The process of recognising aspects of computation and being able to think logically, algorithmically, recursively and inferentially. It typically involves the components of decomposition, pattern recognition, abstraction, modelling and algorithms, which may be used to create digital solutions.
Control structures	Code structures that define the algorithmic behaviour of a software solution. For the purposes of this study design: <ul style="list-style-type: none"> <li>Sequence: All lines of code are executed in the same order they appear.</li> <li>Selection (or branching): Code is executed based on the evaluation of a condition and the data provided. Examples include IF, IF...ELSE, IF...ELSEIF and SWITCH/CASE.</li> <li>Repetition (or iteration): Code is executed repeatedly based on the evaluation of a condition. Examples include pre-test (WHILE loops), post-test (DO...WHILE/REPEAT...UNTIL loops) and repeating a block of code a set number of times based on context and the data provided (FOR loop).</li> </ul>
Conventions	General or commonly accepted ways of working with digital systems or representing data to ensure consistency across a solution. Examples include (but are not limited to): <ul style="list-style-type: none"> <li>how data might be aligned within the cells of a spreadsheet</li> <li>using a consistent axis scale in a chart.</li> </ul> Conventions can be enforced, such as when passwords must have specific characters, or recommended, such as including a subject line in emails.

Critical and creative thinking	The process of using a range of techniques when developing ideas and designs. It involves composing, analysing and evaluating questions, data and decisions. Students explore the use of these strategies when managing the process of solving problems.
Cyber security	A multi-faceted field incorporating aspects of digital systems, organisational practices, threats to data and systems, law, ethics and risk management. Cyber security measures focus on protecting systems and data from a range of threats.
Data analytics	The processes and tools that allow organisations to acquire and/or extract data in various forms, update erroneous or incomplete data, analyse the data to identify trends, relationships and patterns, draw inferences about the data and present findings using visual methods that provide clear and unambiguous conclusions.
Data integrity	A method of describing the overall accuracy, authenticity, correctness, reasonableness, relevance and timeliness of data. Data with high integrity is more reliable and trustworthy, while data without integrity should not be used in calculations or be trusted to make decisions.
Data sources	Data accessed by or provided to a solution by an external file or system. Sources may be unstructured (plain text files) or structured (delimited files or XML formats).  See also Primary data and Secondary data.
Data visualisations	The process of using software tools to present analysed data, such as a graphic representation, usually by combining charts, histograms, graphs, maps and network diagrams, in a visually attractive and informative way. Data visualisations help users explore data to identify patterns and relationships in large amounts of data. Data visualisation tools allow graphic representations to be static or dynamic and can incorporate virtual reality and augmented reality.
Debugging	A process used when developing solutions using a programming language that involves identifying existing errors, fixing the errors and testing to see if the changes made are correct. Debugging occurs during the development of a solution as code is written.
Descriptive statistics	A collection of measures that can be used to provide a summary of a data set. These can be referred to as measures of central tendency (average, median) or measures of variability and spread (maximum, minimum, range, frequency, standard deviation).
Design principles	Accepted characteristics that contribute to the functionality, usability and appearance of solutions. In this study, the principles are related to: <ul style="list-style-type: none"> <li>• functionality (interactivity and navigation)</li> <li>• usability (ease of use and accessibility)</li> <li>• appearance (alignment, balance, contrast, image use, space, text and table formatting).</li> </ul>
Design thinking	A way of thinking critically and creatively to generate and evaluate innovative ideas, and precisely define the preferred solution so it can be created using a digital system. It involves an understanding of the needs of users and of ways of creating solutions that are more efficient or effective than existing ones. When designing, students use both divergent and convergent thinking skills. Divergent thinking supports creativity and the generation of a range of ideas. Convergent thinking supports the selection of a preferred solution and the preparation of accurate and logical plans and instructions to digitally create the solution.
Digital system	Refers to elements such as hardware and software, and their interconnectedness, used to create digital solutions. When digital systems are connected, they form a network.

Dynamic data visualisations	Graphical representations of complex data or information. They allow the exploration of data in an interactive way. In contrast to a static visualisation or chart, a dynamic data visualisation contains data that can change in response to user interaction or the addition of live data.
Effectiveness	A measure of how well solutions, designs, data and information security strategies and development practices function, and the degree to which they achieve their intended purpose. Effectiveness measures for solutions and designs include accessibility, accuracy, attractiveness, clarity, communication of message, completeness, maintainability, readability, relevance, timeliness and usability. Effectiveness measures for data and information security strategies include confidentiality, integrity and availability. Effectiveness measures for development practices include security controls in place, exposure to vulnerability and risk, and legal compliance.
Efficiency	A measure of how much time, cost and effort is applied to achieve intended results. Measures of efficiency in a solution could include the cost of data and file manipulation, its functionality and the speed of processing. Measures of efficiency in a network include its productivity, processing time, operational costs and level of automation.
Encryption	The process of encoding data and information from a plain text format. This is done to protect data and information from being compromised. There are two methods of encryption: symmetric and asymmetric. Symmetric encryption involves the use of a single key for the encryption and decryption of data. Asymmetric encryption involves the use of a public key for the encryption of data and information and a private key for the decryption of the data and information.
Errors (programming)	<p>Issues that arise that prevent a software solution from functioning as expected. Within software development, errors can be categorised as:</p> <ul style="list-style-type: none"> <li>• syntax errors: errors in code that prevent a solution from being compiled and executed as a result of a programming language's syntax not being followed correctly</li> <li>• logic errors: errors in code that result in the solution generating incorrect output, such as calculations or evaluating conditions</li> <li>• run-time errors: errors that arise during program execution that result in the solution crashing if not handled correctly. Examples include overflow, index out of range, type mismatch and divide by zero errors.</li> </ul>
Ethics	Issues that arise that challenge moral standards, principles or expectations and that can impact individuals and/or organisations.
Format	The physical appearance in which data and information can be presented. These include images, graphs, tables, text and web pages. Formats specify characteristics such as presentation style or arrangement, shape and size.
Ideation	A process for generating and developing ideas that follows a cycle from starting with a concept through to developing a design. Ideas can be expressed as text, images and drawings, and in verbal form. Ideation tools can include mood boards, brainstorming, mind maps, sketches and annotations.

Identity and access management	<p>The process of ensuring that the members of an organisation can access only the data, modules and systems required to perform their designated duties. When implemented effectively, this ensures that:</p> <ul style="list-style-type: none"> <li>• individuals cannot access data, modules or systems beyond their needs</li> <li>• staff are not granted administrator or high-level privileges (unless necessary)</li> <li>• there is a decreased exposure to risk and security breaches for the organisation.</li> </ul>
Industry frameworks	<p>Guidelines developed to ensure organisations and governments follow strategies for protecting the security of their data and information with networks and maintain the highest ethical standards. For the purposes of this study, the following frameworks are relevant:</p> <ul style="list-style-type: none"> <li>• Australia's Artificial Intelligence (AI) Ethics Principles</li> <li>• Essential Eight</li> <li>• Information Security Manual (ISM).</li> </ul>
Infographics	<p>Graphical representations of complex data or information that provide an overview of a topic or area of interest. They rely on combining visual elements to communicate data patterns or trends quickly and clearly or information as data visualisations. These include complementary colour schemes, easy-to-read text fonts, headings, multiple graphs, simple charts and statistics.</p>
Legal requirements	<p>Key legislation (acts) that individuals, organisations and governments are expected to comply with in relation to intellectual property and the privacy of data and information. For the purposes of this study, the following acts are relevant:</p> <ul style="list-style-type: none"> <li>• <i>Copyright Act 1968</i> (Cwlth)</li> <li>• <i>Health Records Act 2001</i> (Vic)</li> <li>• <i>Privacy Act 1988</i> (Cwlth), including <i>Privacy Amendment (Enhancing Privacy Protection) Act 2012</i> and <i>Privacy Amendment (Notifiable Data Breaches) Act 2017</i></li> <li>• <i>Privacy and Data Protection Act 2014</i> (Vic).</li> </ul>
Naming conventions	<p>A set of guidelines for providing consistency in the naming of entities during the development of digital solutions. These include program names, function names, interface controls, variable names, table names and file names. Naming conventions can minimise the effort needed to read and understand the solution or code.</p>
Primary data	<p>Data that is collected by researchers directly from sources using methods such as surveys, interviews and observations. The data is raw and has not been processed or summarised.</p>
Project management	<p>Detailed proposal for managing projects considering requirements and constraints. Features of project management include identification of tasks, sequencing, time allocation, dependencies, milestones and critical path.</p>
Pseudocode	<p>A series of English-like statements used to represent an algorithm that will solve a problem. Though there is no standard to pseudocode, there are a number of conventions, including:</p> <ul style="list-style-type: none"> <li>• the use of START/BEGIN and FINISH/END to indicate the beginning and end of a program</li> <li>• IF...ELSE and SELECT to represent decisions/selections</li> <li>• REPEAT...UNTIL, WHILE...DO and FOR...NEXT to represent loops/iteration/repetition.</li> </ul>

Secondary data	Data that has been previously collected, summarised or analysed by someone other than the researcher. These can be large amounts of data and stored in repositories or data sets, and could also include reports and articles.
Security controls	Hardware, software, physical equipment, procedures and electronic measures used to assist in the protection of individuals, systems, networks, organisations and data that is collected, communicated and stored.
Security threats	<p>The actions, devices and events that threaten the integrity and security of data and information stored within, and communicated between, digital systems. The threats can be:</p> <ul style="list-style-type: none"> <li>• accidental: deletion or overwriting of data, misdelivery of information and unintended equipment damage</li> <li>• deliberate: insider threats, unauthorised access, theft of data or physical devices, malware, denial of service attacks and social engineering</li> <li>• events-based: natural disasters and environmental factors, power or network outages, hardware failures and data corruption.</li> </ul>
Software requirements specification (SRS)	The intended purpose and environment of a software solution. It documents the key activities associated with the analysis stage of the problem-solving methodology. Features of an SRS should include a description of the functional and non-functional requirements, constraints, scope, user characteristics and technical environments.
Systems thinking	A holistic approach to the identification and solving of problems. Systems thinking involves analysing the interactions and interrelationships between components of individual information systems (data, processes, people and digital systems) to identify how they are influencing the functioning of the whole system.
User experience (UX)	The consideration of how users perceive and respond when interacting with a digital system. Characteristics of UX include affordance, interoperability, security (authentication and data protection) and usability.
Validation	<p>The checking of data for its reasonableness and completeness by manual or electronic means. Validation of data includes:</p> <ul style="list-style-type: none"> <li>• existence or presence checking, which verifies that a required field has a value entered and is not empty or blank</li> <li>• type checking, which confirms that data entered is of a particular type</li> <li>• range checking, which involves ensuring that data entered falls within a certain range.</li> </ul> <p>Validation may occur as the data is input or once it has been stored.</p>
Verification	The checking of data for accuracy and consistency that occurs after data acquisition/entry and throughout the process of manipulation and analysis. This can include proofreading, confirming the data against other sources and checking that the data visualised is consistent with the data acquired/collected.



# Units 1 to 4: Problem-solving methodology

The problem-solving methodology comprises the four stages of: analysis, design, development and evaluation. For each of these stages there is a typical set of activities, as shown in Figure 1 below. Specific details of the scope of the problem-solving methodology stages are provided in the Problem-solving methodology specifications and the introduction to the relevant areas of study. Note: when creating solutions, this methodology can be applied as a single stage-by-stage problem-solving process or applied to each iteration of an agile problem-solving process.

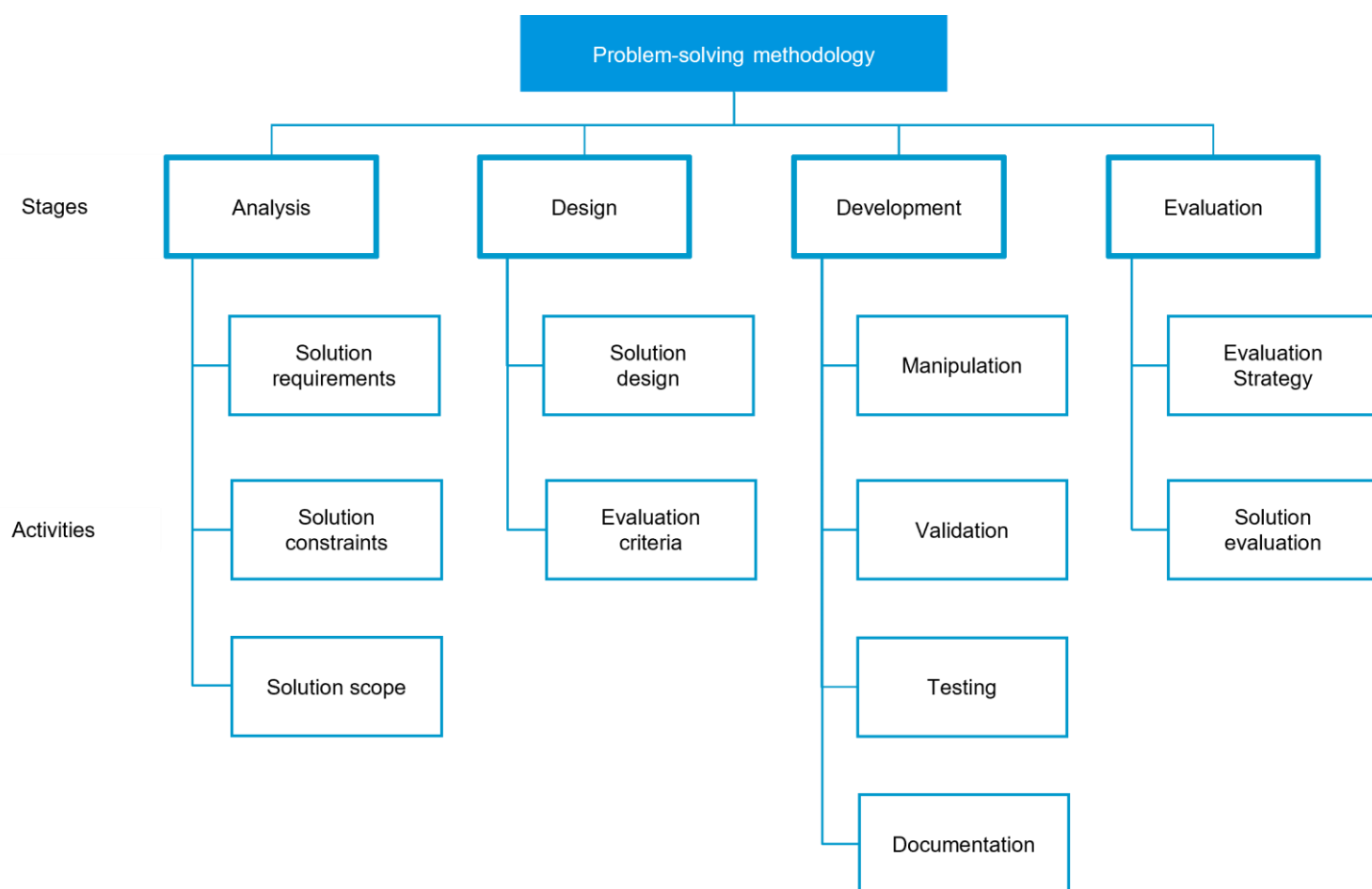


Figure 1: Problem-solving methodology for creating digital solutions

# Units 1 to 4: Problem-solving methodology specifications

Stage and activities	Description	Unit application					
		AC		DA		SD	
		1	2	3	4	3	4
<b>Analysis:</b> involves determining what is required to solve a problem. This involves acquiring and analysing data, and then identifying the solution requirements, constraints and scope.							
Solution requirements	Solution requirements can be described as functional and non-functional.						
	Determine the functional requirements of the solution. These describe what the software solution should do. This involves specific details such as input required, output developed and functions of the solution, including data manipulation and validation.	•	•	•		•	
	Determine the non-functional requirements. These describe the quality attributes of the solution, including usability, reliability, portability, robustness and maintainability.	•	•	•		•	
	Use tools to assist in determining solution requirements, including context diagrams, data flow diagrams and use case diagrams.					•	
Solution constraints	Solution constraints can be described as the conditions or limitations that must be considered when designing a solution.						
	Determine the constraints of the solution. These include economic (cost and time), technical (speed of processing, capacity, availability of equipment, compatibility, security), social (level of expertise of users), legal (intellectual property, ownership of data, privacy of data), and usability (accessibility, usefulness, ease of use).	•	•	•		•	
Solution scope	Solution scope describes the boundaries or parameters of the solution.						
	Identify what will be and/or what will not be addressed by the solution.	•	•	•		•	
Skills underpinning the Analysis stage	Collect data using a range of techniques to determine and clarify user needs, solution requirements, constraints and scope.		•	•		•	
	Collect primary and secondary data and prepare data for identifying trends, patterns and relationships.	•		•			
	Critically analyse the sources of data and information to determine reliability, validity and relevance.	•	•	•		•	
	Draft and evaluate questions to critically analyse requirements, problems, needs or opportunities.	•	•	•		•	
	Develop strategies for asking follow-up questions to further clarify the data and information collected.		•	•		•	

Stage and activities		Description	Unit application					
			AC		DA		SD	
			1	2	3	4	3	4
<b>Design:</b> involves determining how the solution requirements will function and appear. It involves designing the functionality, appearance and user interface of the solution and developing evaluation criteria to determine the efficiency and effectiveness of the software solutions.								
Solution design	Solution design focuses on the development of design ideas into preferred designs using design principles and a range of relevant design tools.							
	Design how the solution will function and appear by identifying the specific data required and how it will be named, structured, validated and manipulated.							
	Use ideation techniques and tools for generating design ideas, including mood boards, brainstorming, mind maps, sketches and annotations.							
	Use design tools to demonstrate structure and functionality, including data dictionaries, input–process–output (IPO) charts, object descriptions, flowcharts and pseudocode.							
	Use design tools to show the relationships between the components of a solution, including annotated diagrams, mock-ups, storyboards, sitemaps, queries, context diagrams, data flow diagrams and use case diagrams.							
	Use design tools to design the visual components of a solution (user interface, graphic representations or data visualisations), including layout diagrams, annotated diagrams and mock-ups. This includes identifying the position and size of text, images and graphics, font types and styles, colours and text enhancements.							
Skills underpinning the Design stage	Investigate existing and possible solutions.							
	Generate design ideas and develop connections between ideas using text and diagrams.							
	Produce detailed designs with annotations, including appearance and functionality of the solution, based on design ideas.							
	Determine possible contingencies when developing solution designs to mitigate issues.							
Evaluation criteria	Evaluation criteria measure the efficiency and effectiveness of designs and the entire solution.							
	Develop evaluation criteria to determine the degree to which solution designs meet requirements.							

		<i>Note that the same evaluation criteria are used in the Evaluation stage to determine the degree to which the solution meets requirements. In the Evaluation stage, the evaluation criteria measure the efficiency and effectiveness of the solution.</i>					
		Modify and refine the evaluation criteria as the solution develops.					

Stage and activities		Description	Unit application						
			AC		DA		SD		
			1	2	3	4	3	4	
<b>Development:</b> involves transforming the requirements and designs into a working software solution. It involves the manipulation and validation of data, testing to ensure the software solution meets requirements and generating documentation to support the use of the solution.									
	Manipulation	Manipulation involves applying a range of functions and techniques using software tools.							
		Develop the solution to meet specifications using appropriate algorithms, features of programming languages or software functions and techniques.	•	•	•	•	•	•	
		Apply appropriate formats and conventions to enhance the appearance of solutions.	•		•	•	•	•	
		Perform calculations using descriptive statistics and identify trends, patterns and relationships.	•		•	•			
	Validation	Validation checks the reasonableness of data being input. Validation can be both manual and electronic.							
		Use validation techniques to check data entry for reasonableness and completeness of data, including existence check, range check and type check.	•		•	•	•	•	
		Use verification techniques after data entry to ensure that data entered matches the source data, including proofreading, confirming the data against other sources and checking that the data visualised is consistent with the data acquired/collected.				•			
		<i>Note that the effectiveness of validation is determined through the testing activity below.</i>							
	Testing	Testing checks whether solutions meet all the requirements, function as expected and are usable by intended users.							
		Conduct debugging techniques to ensure solutions meet requirements.	•				•	•	
		Develop a testing strategy to ensure that the solution works as intended, including features to be tested, test data and expected results.	•	•	•	•	•	•	
		Conduct tests, record the actual results and compare the actual results of the tests against the expected results.	•	•	•	•	•	•	
		Correct identified errors.	•	•	•	•	•	•	

	Documentation	Documentation is used to explain the structure and behaviour of a solution.						
		Write internal documentation to support the functioning, maintenance and upgrading of the solution.	•				•	•
		Write documentation to support the development of solutions.		•				
	Skills underpinning the Development stage	Determine possible contingencies when developing solutions to mitigate issues.		•		•		•
		Modify and evolve designs as the solution develops.		•		•		•
		Modify and refine the evaluation criteria as the solution develops.		•		•		•

Stage and activities		Description	Unit application					
			AC		DA		SD	
			1	2	3	4	3	4
<b>Evaluation:</b> involves determining the degree to which the software solution has met requirements. It involves developing a strategy to evaluate the software solution after implementation and evaluating the software solution against the evaluation criteria to see how well it meets requirements.								
	Evaluation strategy	Evaluation strategies focus on how the solution will be evaluated, including evaluation criteria, time frame and responsibility.						
		Propose a strategy to evaluate the extent to which the solution meets identified problems, needs or opportunities, or solves the identified problem after the solution has been developed and implemented.				•		•
		Specify a timeline for evaluation, the data to be collected, methods and techniques for data collection, who is responsible for collection and how the data relates to the evaluation criteria.				•		•
	Solution evaluation	Solution evaluation determines whether a solution has met requirements.						
		Use the evaluation criteria to determine how the solution has met requirements, needs or opportunities. Discuss in terms of efficiency and effectiveness of the solution.		•		•		•
	Skills underpinning the Evaluation stage	Critically evaluate the analysis process and discuss how it assisted in meeting the requirements, problems, needs or opportunities.				•		•
		Critically evaluate the design process and discuss how it assisted in meeting the requirements, needs or opportunities.				•		•
		Identify and discuss any improvements that could be made to the solution by approaching the problem-solving methodology process differently.				•		•

# Unit 1: Applied computing

In this unit students are introduced to the stages of the problem-solving methodology. Students focus on how data can be used within software tools such as databases and spreadsheets to create data visualisations, and the use of an object-oriented programming (OOP) language to develop a working software solution.

In Area of Study 1, as an introduction to data analytics, students respond to teacher-provided solution requirements, designs and data to develop data visualisations. They develop a solution that includes a database, spreadsheet(s) and data visualisations. In Area of Study 2, students respond to solution requirements to design and develop a working software solution using an OOP language. They develop techniques for debugging and testing their software solution to ensure that it works as intended.

## Software tools

The following table indicates the software tools that students are required to both study and use in this unit.

Area of Study 1	Database software, spreadsheet software and data visualisation software An appropriate tool for running Structured Query Language (SQL) queries
Area of Study 2	An appropriate OOP language

A prescribed list of software tools and functions, and outcome-specific requirements, will be published annually by the VCAA in the [VCAA Bulletin](#). Schools are to select software tools that fulfil these requirements.

## Area of Study 1

### Data analysis

In this area of study, students use software tools to create data visualisations in response to teacher-provided solution requirements, designs and data. The software tools are used for the interpretation and manipulation of data to draw conclusions and create data visualisations that represent their findings. Data visualisations could include charts, graphs, histograms, maps, network diagrams, spatial relationships, infographics and dashboards. No restrictions are placed on the software tools used to create data visualisations and dashboards.

Students examine the features of different design tools to represent the functionality and appearance of software solutions. They interpret given designs and develop a database, spreadsheet(s) and data visualisations using the data provided. Students focus on the appropriate functions and techniques to manipulate and validate data and to make use of suitable formats and conventions.

Students apply computational thinking skills when extracting meaning from data and apply design thinking skills and knowledge to develop data visualisations.

### Outcome 1

On completion of this unit the student should be able to interpret teacher-provided solution requirements and designs, analyse data and develop data visualisations to present findings.

To achieve this outcome the student will draw on key knowledge and key skills outlined in Area of Study 1.



## Key knowledge

- types and purposes of qualitative and quantitative data, such as:
  - interviews and surveys to gather insights/perspectives on a topical issue
  - sensor data to monitor a person's health
  - census and demographic data for statistical analysis
  - data collected over a period of time
  - data generated by artificial intelligence
- characteristics of data types relevant to the selected software tools, such as:
  - text (character, string)
  - numeric (integer, floating point, date/time)
  - Boolean
- factors affecting the quality of data and information, such as:
  - accuracy
  - bias
  - integrity
  - relevance
  - reliability
- characteristics of data and information, such as:
  - size
  - structure
  - relevance
  - accessibility
  - clarity
  - context
- techniques for applying the Australian Privacy Principles (APPs) in the *Privacy Act 1988* (Cwlth) relating to the use, management and communication of data and information, such as:
  - non-identification of individuals (APP 2)
  - information only being held for its primary purpose (APP 6)
  - security measures used to protect personal information (APP 11)
- ethical issues arising from the management and communication of data and information, such as:
  - lack of transparency
  - use of inaccurate or incomplete data
  - ownership and control of data
  - misuse of personal data and information
  - repurposing and sharing of data by artificial intelligence systems
- referencing primary and secondary data and information using the American Psychological Association (APA) referencing system to acknowledge intellectual property
- characteristics of functional and non-functional requirements, constraints and scope
- design tools for representing the functionality and appearance of databases, spreadsheets and data visualisations, such as:
  - input-process-output (IPO) charts
  - annotated diagrams

- mock-ups
- query designs
- structural characteristics of relational database management systems (RDBMS), such as:
  - tables
  - queries
  - relationships using primary and foreign keys
- structural characteristics of spreadsheets, such as:
  - rows and columns
  - cells
- software functions and techniques for efficiently and effectively manipulating, validating and testing data to develop databases, spreadsheets and data visualisations, such as:
  - formulas and functions
  - charts and graphs
  - use of SQL to generate queries
- use of spreadsheets to calculate descriptive statistics for analysis, such as:
  - average
  - median
  - count/frequency
  - standard deviation
- purposes of data visualisations for educating, entertaining, informing and persuading audiences
- types of data visualisations, such as:
  - infographics (long-form)
  - series of posters of infographics
  - dashboards
- components of data visualisations, such as:
  - text and graphics
  - tables
  - charts and graphs
- formats and conventions suitable for databases, spreadsheets and data visualisations, such as:
  - consistent naming conventions for databases (database name, table name, column name, primary key, foreign key)
  - consistent naming conventions for spreadsheets (worksheet names, header labels for rows, header labels for columns)
  - use of colours, fonts, images and icons.

### Key skills

- interpret solution requirements, constraints and scope
- interpret designs using appropriate design tools to represent the functionality and appearance of databases, spreadsheets and data visualisations
- explore data and information from primary and secondary sources, taking into account legal and ethical considerations
- use the APA referencing system to acknowledge intellectual property

- conduct statistical analysis to identify trends and patterns
- select and apply functions, formats, conventions, data validation and testing techniques to efficiently manipulate data and create data visualisations.

## Area of Study 2

### Programming

In this area of study, students use an appropriate OOP language to create a working software solution in response to teacher-provided solution requirements. Students apply the problem-solving stages of design and development to develop the software solution. Details of the relevant problem-solving methodology specifications are on [pages 18–23](#).

Students use a range of design tools to represent the software solution. They develop their working software solution using a range of features of the selected OOP language. The solution should include the use of data structures, naming conventions, internal documentation and validation. Students can choose whether to develop a graphical user interface for the software solution. They apply testing and debugging techniques to ensure the software solution works as intended.

Students apply computational and design thinking skills when preparing solution designs and transforming them into a working solution.

### Outcome 2

On completion of this unit the student should be able to interpret teacher-provided solution requirements to design and develop a software solution using an object-oriented programming language.

To achieve this outcome the student will draw on key knowledge and key skills outlined in Area of Study 2.

### Key knowledge

- emerging trends in programming languages and artificial intelligence-based (AI) code generators for the development of software solutions, such as:
  - low-code development approaches
  - readability and/or maintainability improvements
- characteristics of data types, such as:
  - text (character, string)
  - numeric (integer, floating point, date/time)
  - Boolean
- types of data structures, such as:
  - one-dimensional arrays
  - lists
  - records (varying data types, field index)
- characteristics of functional and non-functional requirements, constraints and scope
- design tools for representing the functionality and appearance of solution designs, such as:
  - mock-ups
  - input-process-output (IPO) charts
  - flowcharts/pseudocode

- key legal requirements relating to intellectual property and copyright while designing and developing software
- principles of OOP, such as:
  - abstraction
  - encapsulation
- features of a programming language, such as:
  - variables, and initialising, accessing and storing data in variables
  - control structures (sequence, selection and iteration/repetition)
  - arithmetic, logical and conditional operators
  - procedures, functions and methods
- naming conventions for solution elements, such as:
  - Hungarian notation
  - camel casing
- purposes of internal documentation, such as:
  - explaining data and code structures
  - code maintenance
- validation techniques for data, such as:
  - existence checking
  - type checking
  - range checking
- debugging and testing techniques for checking software solutions function correctly, such as:
  - test tables to compare expected and actual output
  - construction of relevant test data
  - breakpoints
  - debugging output statements.

### Key skills

- interpret solution requirements to develop a software solution
- select and use appropriate design tools to represent solution designs
- use a range of data types and data structures
- develop a software solution using appropriate features of an OOP language
- document the functioning of a software solution through internal documentation
- design and apply suitable validation, debugging and testing techniques.

## Assessment

The award of satisfactory completion for a unit is based on whether the student has demonstrated the set of outcomes specified for the unit. Teachers should use a variety of learning activities and assessment tasks that provide a range of opportunities for students to demonstrate the key knowledge and key skills in the outcomes.

The areas of study, including the key knowledge and key skills listed for the outcomes, should be used for course design and the development of learning activities and assessment tasks. Assessment must be a part

of the regular teaching and learning program and should be completed mainly in class and within a limited time frame.

All assessments at Units 1 and 2 are school-based. Procedures for assessment of levels of achievement in Units 1 and 2 are a matter for school decision.

For this unit students are required to demonstrate two outcomes. As a set these outcomes encompass the areas of study in the unit.

Suitable tasks for assessment in this unit may be selected from the following:

**Outcome 1**

- A folio of exercises to demonstrate the learning of database, spreadsheet and data visualisation software tools.
- A solution including the use of database, spreadsheet and data visualisation software tools in response to teacher-provided solution requirements and designs.
- A personal portfolio to showcase the development of databases, spreadsheets and data visualisations.

**Outcome 2**

- A folio of exercises to demonstrate the learning of an object-oriented programming language.
- A software solution that includes the designs, solution and a testing table in response to teacher-provided solution requirements.
- The creation and maintenance of code repositories to track the progression of students' learning, using platforms such as GitHub.
- A software solution developed in response to a teacher-provided problem-solving challenge, presented as a hackathon.

Where teachers allow students to choose between tasks, they must ensure that the tasks set are of comparable scope and demand.

## Unit 2: Applied computing

In this unit students focus on developing an innovative solution to a problem, need or opportunity that they have identified, and develop an understanding of network environments, cyber security risks, threats to networks and strategies to reduce the risks to data and information.

In Area of Study 1, students work collaboratively and select a topic of interest involving an emerging trend for further study to create an innovative solution. The innovative solution can be presented as a proof of concept, a prototype or a product. Students engage in all areas of the problem-solving methodology while developing this solution. In Area of Study 2, as an introduction to cyber security, students investigate networks and the threats, vulnerabilities and risks to data and information. They propose and justify strategies to protect the security of data and information within a network.

### Software tools

The following table indicates the software tools that students are required to both study and use in this unit.

Area of Study 1	Any software tools used to design and develop an innovative solution, such as a programming language, spreadsheet software, presentation software or tool for planning a project
-----------------	--

A prescribed list of software tools and functions, and outcome-specific requirements, will be published annually by the VCAA in the [VCAA Bulletin](#). Schools are to select software tools that fulfil these requirements.

## Area of Study 1

### Innovative solutions

In this area of study, students work collaboratively to design and develop an innovative solution to an identified problem, need or opportunity. They apply all stages of the problem-solving methodology to investigate the use of digital devices and emerging digital technologies and their applications. Details of the problem-solving methodology specifications are on [pages 18–23](#).

The innovative solution may take the form of a proof of concept, prototype or product. The final form of the innovative solution can be decided by the students in consultation with their teacher, because the complexity of the innovative solution to address the problem, need or opportunity may not allow for either a working prototype or product to be developed. Since the emphasis is on a collaborative project, students should negotiate a problem, need or opportunity to research and then design and develop their innovative solution.

As the technological landscape continues to rapidly evolve, the potential topics that students can explore will change. For example, students could choose one of the following topics to explore in greater detail:

- artificial intelligence, machine learning or neural networks, such as:
  - generating dashboards through the writing and refinement of prompts
  - using TensorFlow (open source machine learning framework) to analyse and predict trends in data sequences
  - building and training image classification models, for example, to distinguish crown-of-thorns starfish from those starfish that live in harmony with coral reefs

- assistive and wearable technologies or Internet of Things (IoT), such as:
  - communication aids for individuals with speech or motor impairments
  - proximity sensors for individuals with visual impairments to help navigate their surroundings
  - smart braille displays that convert digital text to tactile braille output
- smart devices in healthcare, such as:
  - fall detection systems that automatically send alerts to caregivers or emergency services
  - automated pill dispensers for individuals with dementia
- drones, microcontrollers, nanosatellites and robotic devices, such as:
  - drones or nanosatellites that use microcontroller sensors to collect hygrometry data, which is either stored on an SD card or telemetered (radio transmission) to a receiving ground station (laptop/computer)
- games development or multimedia programming, such as:
  - narrative-driven games that tell compelling stories related to cyberbullying, climate change or Aboriginal and Torres Strait Islander Peoples
  - simulation games that challenge players to construct and manage eco-friendly cities
  - multimedia programming that guides individuals through a fitness regime or mindfulness exercises
- mixed realities, such as augmented or virtual reality, to develop programs that use avatars to deliver educational content, or that create a virtual relaxation space with serene landscapes, soothing sounds and guided mindfulness exercises
- blockchain, such as:
  - solving problems in food supply chains
  - a game to demonstrate how blockchain technology would work in a particular industry, such as the pharmaceutical industry
- innovative uses for emerging technologies, such as:
  - artificial intelligence-based (AI) code generators to build an AI-based debugger that assists in identifying and fixing errors in code
- any other innovative digital solution.

A project plan is prepared to support an organised approach to problem solving. Students use software to record the identification and sequencing of tasks, time allocation, milestones, dependencies and critical path. They record and monitor the progress of their innovative solution throughout the stages of the problem-solving methodology. Students do not have to use dedicated project management software.

Students apply computational, design and systems thinking skills when developing solution designs and transforming them into a proof of concept, prototype or product.

## Outcome 1

On completion of this unit the student should be able to, in collaboration with other students, identify a problem, need or opportunity to analyse, design, develop and evaluate an innovative solution.

To achieve this outcome the student will draw on key knowledge and key skills outlined in Area of Study 1.

## Key knowledge

- the role of curiosity, ingenuity and the United Nations' Sustainable Development Goals to inspire and drive innovation
- functions and capabilities of digital systems used by individuals and organisations, such as:
  - assistive technologies
  - financial services
  - global positioning system (GPS) devices
  - robotics
  - traffic management
- the impact of emerging technologies, such as:
  - automation
  - cyberbullying
  - productivity
  - economic issues (deskilling, job loss, misuse)
  - the decline of physical human interactions and interpersonal skills
- characteristics of creative and innovative solutions, such as:
  - originality
  - effectiveness
  - user-centred
  - disruptiveness
- solution specifications, such as:
  - functional and non-functional requirements
  - constraints
  - scope
- methods for collecting data to determine user needs and requirements, such as:
  - interviews
  - surveys
- design tools and techniques for representing solution designs, such as:
  - mock-ups
  - pseudocode
  - sitemaps
  - storyboards
- techniques for developing innovative solutions, such as:
  - empathising with the end-user
  - ideating for exploring multiple perspectives for a problem
  - generating a divergence of possible design ideas
  - developing the preferred design idea
- techniques for documenting the development of solutions, task delegation and monitoring project progress, such as:
  - Gantt charts
  - project journals
  - version control



- techniques for testing and evaluating innovative solutions, such as:
  - user testing
  - expert review
- how emerging technologies are affected by key legislation and frameworks, such as:
  - Australia’s Artificial Intelligence (AI) Ethics Principles
  - Copyright Act 1968 (Cwlth)
  - Health Records Act 2001 (HPP 1, 2, 5)
  - *Privacy Act 1988* (Cwlth) (APP 1, 2, 6)
  - Privacy and Data Protection Act 2014 (IPP 1, 2, 5, 8)
- ethical issues arising from the development of emerging technologies, such as:
  - cyber security threats
  - biometric systems collecting and storing data
  - job displacement
- ethical issues arising from the use of artificial intelligence, such as:
  - creating content that is biased, discriminatory or otherwise harmful
  - creating content that could be used for cyber attacks
  - generating content from existing copyright materials
- evaluation criteria and techniques for evaluating the efficiency and effectiveness of innovative solutions.

### Key skills

- investigate a problem, need or opportunity and identify potential users and purpose
- propose and apply a range of methods to collect data for analysis
- analyse and document solution requirements to develop an innovative solution
- select and use appropriate design tools for generating solution designs
- develop an innovative solution using appropriate digital systems
- document the development of an innovative solution
- design and apply suitable testing techniques
- identify and discuss potential legal and ethical issues affecting the development of an innovative solution
- apply evaluation criteria and evaluate the efficiency and effectiveness of an innovative solution to meet a problem, need or opportunity.

## Area of Study 2

### Cyber security

The awareness of cyber security incidents and how to be protected from them is more important than ever, especially with the proliferation of mobile devices and online services. In this area of study, students investigate emerging trends in cyber security and how networks enable data and information to be exchanged locally and globally. Students examine the hardware and software components and procedures required to connect, maintain and protect wired, wireless and mobile communications technologies. They apply this knowledge to develop an understanding of cyber security issues when investigating security threats to data and information on networks. Students examine network vulnerabilities and the strategies for reducing risks and mitigating threats to networks, taking into account key legal and ethical requirements. Frameworks such as Australia’s Artificial Intelligence (AI) Ethics Principles and the Essential Eight enable

students to follow current industry practice when considering the use of artificial intelligence and strategies for protecting the security of data and information within networks.

Students apply computational, design and systems thinking skills when analysing networks and proposing strategies for reducing security risks.

## Outcome 2

On completion of this unit the student should be able to respond to a teacher-provided case study to examine a cyber security incident or a network vulnerability, evaluate the threats to a network, and propose strategies to protect the security of data and information on the network.

To achieve this outcome the student will draw on key knowledge and key skills outlined in Area of Study 2.

### Key knowledge

- emerging trends in cyber security, such as:
  - security measures (multi-factor authentication)
  - regulatory compliance
  - artificial intelligence-based (AI) threat detection
  - Zero Trust Architecture
  - use of cryptography for protection strategies
- functions and characteristics of key hardware and software components of networks required for communicating and storing data and information, such as:
  - routers for connecting multiple networks
  - switches for connecting multiple devices in a network
  - firewalls for monitoring and controlling incoming and outgoing network traffic
  - data storage and backup systems
  - network security software (firewall software, antivirus software, Darktrace AI) for protecting networks from threats and vulnerabilities
  - technical underpinnings of intrusion detection systems (IDS) and intrusion prevention systems (IPS)
- strengths and limitations of wired, wireless and mobile communications technology, measured in terms of:
  - cost
  - data storage options
  - reliability
  - security
- technical underpinnings of intranets, the internet and virtual private networks
- applications and capabilities of Local Area Networks (LANs), Wide Area Networks (WANs) and Wireless Personal Area Networks (WPANs)
- risks of using networks in a global environment, such as:
  - cyber security threats
  - data privacy
  - legal compliance
  - unauthorised network access

- technical underpinnings of malware that can intentionally threaten the security of networks, such as:
  - spyware
  - viruses
  - worms
  - ransomware
- security threats to data and information on networks, such as:
  - denial of service attacks
  - improper credential management
  - malicious software
  - outdated versions of software
  - weak passwords
- practices for reducing risks and mitigating threats to networks, such as:
  - application of firmware
  - multifactor authentication
  - backup strategies
  - operating system updates
  - software malware updates
  - staff procedures
- cryptographic techniques to protect data and networks, such as:
  - ciphers (Caesar cipher, Vigenère cipher, polyalphabetic cipher)
  - symmetric encryption (AES)
  - asymmetric encryption (ECDH, ECDSA, RSA)
- emergence of artificial intelligence in providing network security mechanisms, such as:
  - machine learning algorithms that analyse network traffic patterns
  - provision of real-time monitoring and notification
- the role of ethical hacking, such as identifying vulnerabilities and weaknesses in networks
- key legislation and industry frameworks that affect how organisations ethically control the security and communication of data and information:
  - Essential Eight
  - Health Records Act 2001 (HPP 2, 4, 5)
  - Information Security Manual (ISM) (Guidelines for Networking: Encryption; Segmentation and segregation; Network access controls; Confidentiality and integrity of wireless network traffic; Wireless network footprint)
  - *Privacy Act 1988* (Cwlth) (APP 1, 6, 11)
  - Privacy and Data Protection Act 2014 (IPP 2, 4, 5).

## Key skills

- identify and examine a cyber security incident or a network vulnerability
- identify and describe the key components of networks
- describe the capabilities of different networks
- identify and evaluate the impact of network vulnerabilities and threats to the security of data and information
- identify and discuss possible legal and ethical issues arising from ineffective security strategies
- propose and justify strategies to protect the security of data and information within a network.

## Assessment

The award of satisfactory completion for a unit is based on whether the student has demonstrated the set of outcomes specified for the unit. Teachers should use a variety of learning activities and assessment tasks that provide a range of opportunities for students to demonstrate the key knowledge and key skills in the outcomes.

The areas of study, including the key knowledge and key skills listed for the outcomes, should be used for course design and the development of learning activities and assessment tasks. Assessment must be a part of the regular teaching and learning program and should be completed mainly in class and within a limited time frame.

All assessments at Units 1 and 2 are school-based. Procedures for assessment of levels of achievement in Units 1 and 2 are a matter for school decision.

For this unit students are required to demonstrate two outcomes. As a set these outcomes encompass the areas of study in the unit.

Suitable tasks for assessment in this unit may be selected from the following:

### Outcome 1

- An innovative solution that includes an analysis, designs, the development of a proof of concept/prototype/product and an evaluation.
- A presentation (oral, multimedia, visual) of an innovative solution.
- A written report that documents the development of an innovative solution.
- An annotated visual report that documents the development of an innovative solution.

### Outcome 2

- A teacher-provided case study with structured questions that investigates a cyber security incident and how it could be prevented in the future.
- A teacher-provided case study with structured questions that investigates a network, its vulnerabilities and how these could be mitigated.

Where teachers allow students to choose between tasks, they must ensure that the tasks set are of comparable scope and demand.

## Unit 3: Data analytics

In this unit students apply the problem-solving methodology to analyse data using software tools such as database, spreadsheet and data visualisation software to create data visualisations. Students develop an understanding of the analysis, design and development stages of the problem-solving methodology.

In Area of Study 1, students respond to teacher-provided solution requirements and designs to develop data visualisations. They apply specific functions of database and spreadsheet software tools to manipulate, cleanse and analyse data. Students then use a data visualisation software tool to develop data visualisations that present their findings. In Area of Study 2, students propose a research question, prepare a project plan, collect, analyse and prepare data, and design infographics and/or dynamic data visualisations. Area of Study 2 forms the first part of the School-assessed Task (SAT) that is completed in Unit 4, Area of Study 1.

### Software tools

The following table indicates the software tools that students are required to both study and use in this unit.

Area of Study 1	Database software, spreadsheet software and data visualisation software An appropriate tool for running Structured Query Language (SQL) queries
-----------------	--

The following table indicates the software tools that students are required to use, but are not required to study, in this unit.

Area of Study 2	An appropriate tool for documenting and modifying project plans Appropriate tools for ideation and generating designs
-----------------	--

A prescribed list of software tools and functions, and outcome-specific requirements, will be published annually by the VCAA in the [VCAA Bulletin](#). Schools are to select software tools that fulfil these requirements.

## Area of Study 1

### Data analytics

In this area of study, students identify, select and extract authentic data from large repositories. They manipulate the data to present findings as data visualisations in response to teacher-provided solution requirements and designs. Students use database, spreadsheet and data visualisation software tools to undertake the problem-solving activities of manipulation, validation and testing in the development stage.

The software solutions involve importing data from files into a database where data is organised using queries. Data from these queries is then imported into a spreadsheet for further refinement to identify trends, relationships and patterns before presenting findings as data visualisations. This will prepare students for creating infographics and/or dynamic data visualisations in Unit 4, Area of Study 1. Students are not required to manually calculate statistics. They are expected to use relevant spreadsheet functions to calculate values. Validation and testing techniques are applied to ensure the reasonableness and completeness of the data used to develop data visualisations. Students justify the use of functions, formats and conventions in the development of their data visualisations.

Students apply computational thinking skills when interpreting solution requirements and designs, and when developing them into data visualisations.

## Outcome 1

On completion of this unit the student should be able to interpret teacher-provided solution requirements and designs, extract data from large repositories, manipulate and cleanse data, conduct statistical analysis and develop data visualisations to display findings.

To achieve this outcome the student will draw on key knowledge and key skills outlined in Area of Study 1.

### Key knowledge

- emerging trends in data analytics using artificial intelligence, including:
  - integration of artificial intelligence features into software tools
  - generating data visualisations through the writing and refinement of prompts
  - machine learning and statistical modelling for making predictions, decisions and recommendations
- characteristics of functional and non-functional requirements, constraints and scope
- characteristics of data types, including:
  - text (character, string)
  - numeric (integer, floating point, date/time)
  - Boolean
- structural characteristics of relational database management systems (RDBMS), including:
  - data types and field sizes
  - data in tables
  - relationships using primary and foreign key fields
  - use of SQL to generate queries
- design tools for representing databases and spreadsheets, including:
  - data dictionaries
  - query designs
  - layout diagrams
  - input-process-output (IPO) charts
- techniques for identifying, selecting, extracting and validating authentic data stored in large repositories, including:
  - downloading datasets in a range of formats
  - the use of SQL functions to retrieve, filter, sort and link dataset values (SELECT, FROM, WHERE, ORDER BY, INNER JOIN)
  - the use of Boolean operators (AND, NOT, OR) for WHERE statements
  - existence checking, type checking and range checking
- methods for referencing data sources using the American Psychological Association (APA) referencing system
- techniques for effectively and efficiently manipulating and cleansing data, including:
  - formulas and functions to perform calculations
  - sorting, filtering and reformatting
  - identifying and fixing errors

- techniques to statistically analyse data to identify trends, relationships and patterns, including:
  - descriptive statistics (average, median, minimum, maximum, range, standard deviation, count/frequency, sum)
  - Pearson's correlation co-efficient ( $r$ )
  - the shape and skew of data
- purposes of data visualisations, including:
  - exploratory data analysis
  - presentation of information
  - providing interactive experiences for users to explore data
- types of data visualisations, including:
  - infographics (series or long-form, static)
  - dashboards (interactive, static or live data)
  - dynamic data visualisations (interactive, live data)
- design tools for representing data visualisations, including:
  - mock-ups
  - storyboards
- formats and conventions applied to data visualisations to improve their effectiveness for intended users, including:
  - use of colours, fonts, images and icons
  - visual hierarchy and clarity of message
- techniques for testing databases and spreadsheets, including:
  - testing formula and query results
  - testing validation
  - test cases comparing expected and actual results in testing tables
- techniques for testing data visualisations, including:
  - visual inspection of the appearance of the data visualisation
  - confirming that charts and graphs are representative of the data being visualised.

## Key skills

- interpret solution requirements and designs
- identify, select, extract and validate relevant data from large repositories using database software
- use the APA referencing system to acknowledge intellectual property
- manipulate and cleanse data using spreadsheet software
- conduct statistical analysis to identify trends, relationships and patterns
- select, justify and apply functions, formats and conventions to create effective data visualisations
- develop and apply suitable testing techniques to software tools used.

## Area of Study 2

### Data analytics: analysis and design

In this area of study, students independently determine and propose a research question and collect and analyse data from both primary and secondary sources. This is the first part of the School-assessed Task, involving analysis and design, with the second part undertaken in Unit 4, Area of Study 1.

Students prepare a project plan that includes both student-determined and teacher-provided milestones in a Gantt chart that includes all stages of the problem-solving methodology covered in Unit 3, Area of Study 2 and in Unit 4, Area of Study 1. Details of the relevant problem-solving methodology specifications are on [pages 18–23](#). Throughout Unit 3, Area of Study 2 and Unit 4, Area of Study 1, students monitor and modify their project plans as required. They do not have to use dedicated project management software.

A range of methods is used to collect data to determine solution requirements, constraints and scope. Primary and secondary data is then collected to address the research question using appropriate methods and considering legal requirements. The scope of the research question should be sufficiently detailed to not require a simple yes or no answer, but rather an exploration of the data collected.

Students generate and document two to three design ideas for creating the infographics and/or dynamic data visualisations using ideation tools. Evaluation criteria are developed by the students to determine which of the ideas will be used as the basis of the preferred design. These ideas are then fully developed into detailed designs using a range of design tools addressing the appearance, usability and functionality of the solution. These evaluation criteria will be used in Unit 4, Area of Study 1 to evaluate the efficiency and effectiveness of the infographics and/or dynamic data visualisations.

Students apply computational thinking skills when determining data requirements associated with a research question and apply design thinking skills when designing infographics and/or dynamic data visualisations.

### Outcome 2

On completion of this unit the student should be able to propose a research question, formulate a project plan, collect and prepare data, and generate design ideas and a preferred design for creating infographics and/or dynamic data visualisations.

To achieve this outcome the student will draw on key knowledge and key skills outlined in Area of Study 2.

#### Key knowledge

- features of a research question, including:
  - clarity
  - measurability of data requirements
  - feasibility
  - originality
- features of project management to develop a project plan using Gantt charts, including:
  - identification of tasks
  - sequencing of tasks
  - time allocation
  - dependencies
  - milestones
  - critical path
  - monitoring and documenting the progress of projects



- characteristics of functional and non-functional requirements, constraints and scope, including data to address the research question
- methods for collecting primary data, including:
  - surveys
  - interviews
  - observations
- methods for collecting secondary data, including:
  - querying of data stored in large repositories
  - online searches
- characteristics of data types and data structures relevant to manipulating collected data
- suitability of quantitative and qualitative data for manipulation to prepare for data visualisations, including:
  - coding of qualitative data
  - identifying trends, relationships and patterns
- characteristics of data integrity, including:
  - accuracy
  - authenticity
  - correctness
  - reasonableness
  - relevance
  - timeliness
- procedures and techniques for the ethical collection of primary data, including:
  - using participant information statements and/or consent forms
  - de-identifying personal data
- key legal requirements for the protection of intellectual property and the collection, communication and security of data and information, including:
  - Copyright Act 1968 (Cwlth)
  - Health Records Act 2001 (HPP 1, 2, 4)
  - *Privacy Act 1988* (Cwlth) (APP 3, 4, 5, 6, 11)
  - Privacy and Data Protection Act 2014 (IPP 1, 2, 4, 10)
- methods for referencing secondary sources using the APA referencing system to acknowledge intellectual property, including:
  - use of citations
  - creation of reference lists
- procedures and techniques for managing data, including:
  - archiving
  - backups (full, incremental, differential)
  - disposal
- ideation techniques and tools for generating design ideas, including:
  - mood boards
  - brainstorming
  - mind maps

- sketches
- annotations
- design principles that influence the appearance and functionality of infographics and dynamic data visualisations, including:
  - alignment
  - balance
  - contrast
  - image use
  - space
  - text and table formatting
  - usability
  - navigation
  - interactivity
- criteria for evaluating design ideas and the efficiency and effectiveness of infographics and dynamic data visualisations
- design tools for generating solution designs from design ideas, including:
  - storyboards
  - mock-ups
  - input-process-output (IPO) charts
  - query designs.

### Key skills

- propose a research question
- create, monitor and modify project plans using software
- analyse and document solution requirements, constraints and scope of infographics and/or dynamic data visualisations
- apply techniques for searching, collecting, referencing and managing data sets
- generate design ideas using appropriate ideation techniques and tools
- develop evaluation criteria for design ideas and the efficiency and effectiveness of infographics and dynamic data visualisations
- produce detailed designs using appropriate design principles and tools.

## School-based assessment

### Satisfactory completion

The award of satisfactory completion for a unit is based on whether the student has demonstrated the set of outcomes specified for the unit. Teachers should use a variety of learning activities and assessment tasks to provide a range of opportunities for students to demonstrate the key knowledge and key skills in the outcomes.

The areas of study and key knowledge and key skills listed for the outcomes should be used for course design and the development of learning activities and assessment tasks.

## Assessment of levels of achievement

The student's level of achievement in Unit 3 will be determined by School-assessed Coursework and a School-assessed Task.

School-assessed Coursework tasks must be a part of the regular teaching and learning program and must not unduly add to the workload associated with that program. They must be completed mainly in class and within a limited time frame.

Where teachers provide a range of options for the same School-assessed Coursework task, they should ensure that the options are of comparable scope and demand.

The types and range of forms of School-assessed Coursework for the outcomes are prescribed within the study design. The VCAA publishes [Support materials](#) for this study, which include advice on the design of assessment tasks and the assessment of student work for a level of achievement.

Teachers will provide to the VCAA a numerical score representing an assessment of the student's level of achievement. The score must be based on the teacher's assessment of the performance of each student on the tasks set out in the following table.

## Contribution to final assessment

School-assessed Coursework for Unit 3 will contribute 10 per cent to the study score.

Outcomes	Marks allocated	Assessment tasks
<b>Unit 3</b> <b>Outcome 1</b> Interpret teacher-provided solution requirements and designs, extract data from large repositories, manipulate and cleanse data, conduct statistical analysis and develop data visualisations to display findings.	100	In response to teacher-provided solution requirements and designs: <ul style="list-style-type: none"> <li>extract and reference data from large repositories into a database</li> <li>query data using databases and SQL</li> <li>use spreadsheet functions to manipulate data</li> <li>statistically analyse data in spreadsheets</li> <li>develop data visualisations.</li> </ul> Task time allocated should be at least 6–10 lessons.
<b>Total marks</b>	<b>100</b>	

## School-assessed Task

The student's level of achievement in Unit 3, Outcome 2 and in Unit 4, Outcome 1 will be assessed through a School-assessed Task. Details of the School-assessed Task for Units 3 and 4 are provided on [page 51](#) of this study design.

## External assessment

The level of achievement for Units 3 and 4 is also assessed by an end-of-year examination (see [page 52](#)), which will contribute 50 per cent to the study score.

## Unit 4: Data analytics

In this unit students focus on determining the findings of a research question by developing infographics and/or dynamic data visualisations based on large complex data sets, consider data breaches and investigate the security strategies used by an organisation to protect data and information from cyber security threats.

In Area of Study 1, students apply the problem-solving stages of development and evaluation to develop their preferred designs prepared in Unit 3, Area of Study 2 into infographics and/or dynamic data visualisations. They evaluate the infographics and/or dynamic data visualisations and assess the project plan. Area of Study 1 forms the second part of the School-assessed Task (SAT). In Area of Study 2, students analyse a case study that investigates the impact of a data breach on an organisation. They examine the cyber security threats to data and information, evaluate security strategies and recommend improved strategies for protecting data and information.

### Software tools

The following table indicates the software tools that students are required to both study and use in this unit.

Area of Study 1	Database software, spreadsheet software and data visualisation software An appropriate tool for running Structured Query Language (SQL) queries
-----------------	--

The following table indicates the software tool that students are required to use, but are not required to study, in this unit.

Area of Study 1	An appropriate tool for documenting and modifying project plans
-----------------	---

A prescribed list of software tools and functions, and outcome-specific requirements, will be published annually by the VCAA in the [VCAA Bulletin](#). Schools are to select software tools that fulfil these requirements.

## Area of Study 1

### Data analytics: development and evaluation

In this area of study, students develop the design they prepared in Unit 3, Area of Study 2 into infographics and/or dynamic data visualisations that address a research question by applying the problem-solving methodology stages of development and evaluation. Details of the problem-solving methodology specifications are on [pages 18–23](#).

Effective designs and clarity of messages are key features of infographics and dynamic data visualisations, which are designed to present findings intended for a target audience. Students import data from files into a database and then into a spreadsheet for refinement. They analyse the data using descriptive statistics in spreadsheets and use data visualisation software to transform the preferred designs into infographics and/or dynamic data visualisations.

Students monitor and record the progress of their projects using the project plan developed in Unit 3, Area of Study 2. Details could include actual versus expected durations, achievement of milestones and annotations to explain progress.

Students evaluate the quality of their infographics and/or dynamic data visualisations using the evaluation criteria developed in Unit 3, Area of Study 2 and assess the effectiveness of their project plan in the completion of their project.

Students apply computational thinking skills when developing their ideas into infographics and/or dynamic data visualisations.

## Outcome 1

On completion of this unit the student should be able to develop and evaluate infographics and/or dynamic data visualisations that meet requirements and assess the effectiveness of the project plan.

To achieve this outcome the student will draw on key knowledge and key skills outlined in Area of Study 1.

### Key knowledge

- effective and efficient methods to manipulate data using software tools, including:
  - use of templates
  - software functions
  - use of artificial intelligence tools to represent data and information
- techniques for analysing data to refine findings for data visualisations, including:
  - descriptive statistics (average, median, minimum, maximum, range, standard deviation, count/frequency, sum)
  - Pearson's correlation co-efficient ( $r$ )
  - the shape and skew of data
- techniques for implementing data security, including:
  - using security controls
  - developing an implementation plan
  - using a checklist to ensure controls are successfully implemented
- characteristics of information for target audiences, including:
  - age appropriateness
  - commonality of language
  - culture and gender inclusiveness
- techniques for developing infographics and dynamic data visualisations, including:
  - organising the hierarchy of information
  - pagination
  - clarifying data refresh requirements
- formats and conventions applied to infographics and dynamic data visualisations to improve their effectiveness for intended users, including:
  - use of colours, fonts, images and icons
  - visual hierarchy and clarity of message
- techniques for improving data quality by validating and verifying data, including:
  - existence checking
  - type checking
  - range checking
  - checking end-to-end data integrity from collection to visualisation

- proofreading
- techniques for testing that data visualisations work as intended, including:
  - visual inspection of the appearance of the data visualisation
  - confirming that charts and graphs are representative of the data being visualised
- features of evaluation strategies, including:
  - criteria
  - time frame
  - responsibility
- techniques for applying evaluation criteria
- techniques for recording the progress of projects, including:
  - adjustments to tasks
  - adjustments to time frames
  - annotations to project plans
  - monitoring and documenting progress using logs/journals
- techniques for assessing the effectiveness of a project plan, including:
  - reviewing the number of changes made to the project plan during the project
  - the reason changes were necessary
  - the impact of changes on the completion of the project.

### Key skills

- monitor, modify and annotate project plans as necessary
- conduct statistical analysis to identify trends, relationships and patterns
- implement procedures for securing data
- apply formats and conventions to develop infographics and/or dynamic data visualisations using software
- select and apply data validation, verification and testing techniques
- evaluate the efficiency and effectiveness of infographics and/or dynamic data visualisations
- assess the effectiveness of the project plan.

## Area of Study 2

### Cyber security: data security

Major data breaches involving large Australian companies and businesses have increased over the last few years. These have impacted millions of Australians. In this area of study, students investigate emerging trends in cyber security and focus on data and information security and its importance to organisations. Students investigate security strategies used by an organisation to manage the collection, communication and security of data and information in its networked environment. They examine the threats to this data and information, and evaluate the methods an organisation uses to protect its data and information. Students consider the consequences for an organisation that fails to protect its data and information. They recommend strategies to reduce the threats to data and information, taking into account the key legal requirements and any ethical issues faced by the organisation, including those posed by the use of artificial intelligence.

Students apply systems thinking skills when investigating data and information security strategies within an organisation, and when recommending strategies to reduce threats.

## Outcome 2

On completion of this unit the student should be able to respond to a teacher-provided case study to analyse the impact of a data breach on an organisation, identify and evaluate threats, evaluate current security strategies and make recommendations to improve security strategies.

To achieve this outcome the student will draw on key knowledge and key skills outlined in Area of Study 2.

### Key knowledge

- emerging trends in cyber security, including:
  - the use of artificial intelligence to protect data
  - authentication procedures
  - threat detection and response
  - analytics of user behaviour
  - privacy protection
- goals and objectives of medium and large organisations
- the importance of data and information security to organisations, including:
  - safeguarding business operations
  - mitigating financial loss and reputational damage
  - compliance with legal obligations
- types of threats to the integrity and security of data and information used by organisations, including:
  - accidental
  - deliberate
  - events-based
- characteristics of data integrity, including:
  - accuracy
  - authenticity
  - correctness
  - reasonableness
  - relevance
  - timeliness
- consequences of diminished data integrity, including:
  - loss of reputation
  - poor decision-making
  - financial loss
  - legal issues
  - impacted operations
- criteria for evaluating the effectiveness of data and information security strategies, including:
  - confidentiality
  - integrity
  - availability

- key legislation that affects how organisations control the collection, communication and security of their data and information, including:
  - Health Records Act 2001 (HPP 1, 2, 4, 5)
  - *Privacy Act 1988* (Cwlth) (APP 1, 3, 4, 5, 6, 7, 11)
  - Privacy and Data Protection Act 2014 (IPP 1, 2, 4, 5, 10)
- key legislation that instructs an organisation to notify impacted individuals and organisations in the event of an eligible data breach likely to result in serious harm, including:
  - Privacy Amendment (Notifiable Data Breaches) Act 2017 (Sections 26WE, 26WF, 26WH, 26WK, 26WL, 26WR)
- ethical issues arising from the implementation of data and information security practices, including:
  - data collection and use
  - identity theft
  - lack of transparency in the event of a cyber security incident
- applications of cryptographic techniques to protect data at rest and in transit, including:
  - use of ciphers
  - symmetric and asymmetric keys
  - use of hashing functions
- features of disaster recovery plans, including:
  - evacuation
  - backing up
  - restoration
  - communication
  - testing
- security controls for preventing and tracking unauthorised access to data and information and minimising data loss.

### Key skills

- analyse a data breach and describe its impact on an organisation
- identify and evaluate the threats to the security of data and information
- examine and describe an organisation's current data and information security strategies
- propose and apply criteria to evaluate the effectiveness of current data and information security strategies
- identify and discuss possible legal and ethical consequences of ineffective data and information security strategies
- evaluate the disaster recovery plan for an organisation
- recommend and justify improvements to current data and information security strategies.



## School-based assessment

### Satisfactory completion

The award of satisfactory completion for a unit is based on whether the student has demonstrated the set of outcomes specified for the unit. Teachers should use a variety of learning activities and assessment tasks to provide a range of opportunities for students to demonstrate the key knowledge and key skills in the outcomes.

The areas of study and key knowledge and key skills listed for the outcomes should be used for course design and the development of learning activities and assessment tasks.

### Assessment of levels of achievement

The student's level of achievement in Unit 4 will be determined by School-assessed Coursework and a School-assessed Task.

School-assessed Coursework tasks must be a part of the regular teaching and learning program and must not unduly add to the workload associated with that program. They must be completed mainly in class and within a limited time frame.

Where teachers provide a range of options for the same School-assessed Coursework task, they should ensure that the options are of comparable scope and demand.

The types and range of forms of School-assessed Coursework for the outcomes are prescribed within the study design. The VCAA publishes [Support materials](#) for this study, which include advice on the design of assessment tasks and the assessment of student work for a level of achievement.

Teachers will provide to the VCAA a numerical score representing an assessment of the student's level of achievement. The score must be based on the teacher's assessment of the performance of each student on the tasks set out in the following table.

## Contribution to final assessment

School-assessed Coursework for Unit 4 will contribute 10 per cent to the study score.

Outcomes	Marks allocated	Assessment tasks
<b>Unit 4</b> <b>Outcome 2</b> Respond to a teacher-provided case study to analyse the impact of a data breach on an organisation, identify and evaluate threats, evaluate current security strategies and make recommendations to improve security strategies.	100	<p>The student's performance will be assessed using one of the following:</p> <ul style="list-style-type: none"> <li>structured questions</li> <li>a report in written format</li> <li>a report in multimedia format.</li> </ul> <p>The case study scenario needs to enable:</p> <ul style="list-style-type: none"> <li>an analysis of the breach</li> <li>an evaluation of the threats</li> <li>recommendations to improve security strategies.</li> </ul> <p>Task time allocated should be 100–120 minutes.</p>
<b>Total marks</b>	<b>100</b>	

## School-assessed Task

The student's level of achievement in Unit 3, Outcome 2 and in Unit 4, Outcome 1 will be assessed through a School-assessed Task.

The School-assessed Task contributes 30 per cent to the study score.

Outcomes	Assessment tasks
<b>Unit 3</b> <b>Outcome 2</b> Propose a research question, formulate a project plan, collect and prepare data, and generate design ideas and a preferred design for creating infographics and/or dynamic data visualisations.	A documented research question and a project plan (Gantt chart) indicating tasks, times, milestones, dependencies and the critical path <b>AND</b> An analysis that defines the requirements, constraints and scope of infographics and/or dynamic data visualisations <b>AND</b> A collection of complex data sets that has been referenced <b>AND</b> A folio of design ideas and evaluation criteria <b>AND</b> Detailed design specifications of the preferred design. Time allocated should be at least 8–10 weeks of class time.
<b>Unit 4</b> <b>Outcome 1</b> Develop and evaluate infographics and/or dynamic data visualisations that meet requirements and assess the effectiveness of the project plan.	Infographics and/or dynamic data visualisations that present findings in response to a research question <b>AND</b> <ul style="list-style-type: none"> <li>an evaluation of the efficiency and effectiveness of infographics and/or dynamic data visualisations</li> <li>an assessment of the effectiveness of the project plan (Gantt chart) in monitoring project progress</li> </ul> in one of the following: <ul style="list-style-type: none"> <li>a written report</li> <li>an annotated visual plan.</li> </ul> Time allocated should be at least 8 weeks of class time.

## External assessment

The level of achievement for Units 3 and 4 is also assessed by an end-of-year examination.

## End-of-year examination

### Contribution to final assessment

The examination will contribute 50 per cent to the study score.

### Description

The examination will be set by a panel appointed by the VCAA. All the key knowledge and key skills that underpin the outcomes in Units 3 and 4 are examinable.

### Conditions

The examination will be completed under the following conditions:

- Duration: 2 hours
- Date: end-of-year, on a date to be published annually by the VCAA
- VCAA examination rules will apply. Details of these rules are published annually in the [VCE Administrative Handbook](#).
- The examination will be marked by assessors appointed by the VCAA.

### Further advice

The VCAA publishes specifications for all VCE examinations on the VCAA website. Examination specifications include details about the sections of the examination, their weighting, the question format(s) and any other essential information. The specifications are published in the first year of implementation of the revised Unit 3 and 4 sequence, together with any sample material.

## Unit 3: Software development

In this unit students apply the problem-solving methodology to develop working software modules using an object-oriented programming (OOP) language. Students develop an understanding of the analysis, design and development stages of the problem-solving methodology.

In Area of Study 1, students respond to teacher-provided solution requirements and designs to develop a set of working software modules through the use of an OOP language. Students examine a set of requirements and a range of software design tools in order to apply specific features of an OOP language to create working software modules. In Area of Study 2, students analyse an identified problem, need or opportunity, prepare a project plan, develop a software requirements specification and design a software solution. Area of Study 2 forms the first part of the School-assessed Task (SAT) that is completed in Unit 4, Area of Study 1.

### Software tools

The following table indicates the software tools that students are required to both study and use in this unit.

Area of Study 1	An appropriate OOP language
Area of Study 2	Unified Modelling Language (UML) tools to create use case diagrams

The following table indicates the software tools that students are required to use, but are not required to study, in this unit.

Area of Study 2	An appropriate tool for documenting and modifying project plans Appropriate tools for ideation and generating designs
-----------------	--

A prescribed list of software tools and functions, and outcome-specific requirements, will be published annually by the VCAA in the [VCAA Bulletin](#). Schools are to select software tools that fulfil these requirements.

## Area of Study 1

### Software development: programming

In this area of study, students interpret the requirements and designs for developing working software modules. Students use a common OOP language and undertake the problem-solving activities of manipulation (programming), validation, testing and documentation in the development stage.

The working software modules should focus on a range of features within the OOP language. Students are expected to fully develop the working software modules in accordance with the given requirements and designs. This includes the development of a graphical user interface for at least two of the four modules. The working software modules will prepare students for creating a complete software solution in Unit 4, Area of Study 1. Validation is applied within relevant modules to ensure input data can be accepted and processed. Debugging and testing techniques are applied to all software modules to ensure they operate as intended and internal documentation is written to explain the functionality of each module. Students justify the use of the selected features of the OOP language and algorithms in the development of their working software modules.

Students apply computational thinking skills when interpreting given solution requirements and designs, and when developing them into the working software modules.

## Outcome 1

On completion of this unit the student should be able to interpret teacher-provided solution requirements and designs and use appropriate features of an object-oriented programming language to develop working software modules.

To achieve this outcome the student will draw on key knowledge and key skills outlined in Area of Study 1.

### Key knowledge

- emerging trends in programming using artificial intelligence, including:
  - using prompts to generate code
  - automated debugging and testing of modules
  - code optimisation
  - responsible and ethical use of artificial intelligence tools
- characteristics of functional and non-functional requirements, constraints and scope
- design tools for representing modules, including:
  - data dictionaries
  - mock-ups
  - object descriptions
  - input-process-output (IPO) charts
  - pseudocode
- characteristics of data types, including:
  - text (character, string)
  - numeric (integer, floating point, date/time)
  - Boolean
- characteristics of data structures, including:
  - one-dimensional arrays
  - two-dimensional arrays
  - records (varying data types, field index)
- characteristics of data sources (plain text (TXT), delimited (CSV) and XML files), including:
  - structure
  - reasons for use
- principles of OOP, including:
  - abstraction
  - encapsulation
  - generalisation
  - inheritance

- features of a programming language, including:
  - local and global variables, and constants
  - data types
  - instructions and control structures (sequence, selection, iteration/repetition)
  - arithmetic, logical and conditional operators
  - graphical user interfaces (GUIs)
  - functions and methods
  - classes and objects
- purposes and features of naming conventions for solution elements (variables, interface controls, code structures), including:
  - Hungarian notation
  - camel casing
  - snake casing
- validation techniques for data, including:
  - existence checking
  - type checking
  - range checking
- purposes of internal documentation, including:
  - explaining and justifying data and code structures
  - code maintenance
  - placeholder comments for future development (stubs)
- algorithms for sorting and searching, including:
  - selection sort
  - quick sort
  - binary search
  - linear search
- types of errors, including:
  - syntax
  - logic
  - runtime (overflow, index out of range, type mismatch, divide by zero)
- debugging and testing techniques for checking modules function correctly, including:
  - use of breakpoints
  - use of debugging statements
  - construction of relevant test data
  - test cases comparing expected and actual output in testing tables.

## Key skills

- interpret solution requirements and designs
- use a range of data types, data structures and data sources
- use and justify appropriate features of an OOP language to develop working software modules
- develop and apply suitable naming conventions and validation techniques within modules
- document the functioning of modules using internal documentation
- develop and apply suitable debugging and testing techniques using appropriate test data.

## Area of Study 2

### Software development: analysis and design

In this area of study, students prepare for the development of a software solution that meets a student-identified problem, need or opportunity. This is the first part of the School-assessed Task, involving analysis and design, with the second part undertaken in Unit 4, Area of Study 1.

Students are expected to independently identify a problem, need or opportunity for developing a software solution from within their community. They prepare a project plan that includes both student-determined and teacher-provided milestones in a Gantt chart that includes all stages of the problem-solving methodology covered in Unit 3, Area of Study 2 and in Unit 4, Area of Study 1. Details of the relevant problem-solving methodology specifications are on [pages 18–23](#). Throughout Unit 3, Area of Study 2 and Unit 4, Area of Study 1, students monitor and modify their project plans as required. They do not have to use dedicated project management software.

A range of methods are used to collect data for analysis. Analysis tools and techniques are used to depict the relationships between users, data and systems and to document the solution requirements, constraints and scope as part of a software requirements specification.

Students generate and document two to three design ideas for creating their solution using ideation tools. Evaluation criteria are developed by the students to determine which of the ideas will be used as the basis of the preferred design. These ideas are then fully developed into detailed designs using a range of design tools, addressing the appearance, structure and functionality of the solution. The same evaluation criteria are then used in Unit 4, Area of Study 1 to evaluate the efficiency and effectiveness of the software solution.

Students apply computational thinking skills when analysing a problem, need or opportunity and apply design thinking skills when designing the solution.

### Outcome 2

On completion of this unit the student should be able to document a problem, need or opportunity, formulate a project plan, document an analysis, and generate design ideas and a preferred design for creating a software solution.

To achieve this outcome the student will draw on key knowledge and key skills outlined in Area of Study 2.

### Key knowledge

- reasons why individuals and organisations undertake software development projects, including:
  - increasing productivity and efficiency
  - reducing costs
  - identifying opportunities to address gaps in the market
  - meeting organisational objectives or needs
- features of a brief that documents a problem, need or opportunity, including:
  - problem/need/opportunity outline
  - proposed users
  - programming languages to be used
  - feasibility
  - originality



- features of project management using Gantt charts, including:
  - identification of tasks
  - sequencing of tasks
  - time allocation
  - dependencies
  - milestones
  - critical path
  - monitoring and documenting the progress of projects
- methods for collecting data to determine needs and requirements, including:
  - interviews
  - observations
  - surveys
  - reports
- characteristics of functional and non-functional requirements
- constraints that influence solution development, including:
  - economic
  - legal
  - social
  - technical considerations
- characteristics of solution scope, including:
  - version/solution boundaries
- analytical tools for depicting the relationships between users, data and systems, including:
  - context diagrams (Level 0) with the components of a system, and entities and data flows
  - data flow diagrams (Level 1) with the components of processes, entities, data stores and data flows
  - use case diagrams with the components of a system boundary, actors, associations, relationships (includes and extends) and use cases
- purpose and features of software requirements specifications, including:
  - defining requirements
  - constraints
  - scope
  - user characteristics
  - technical environments
  - analytical tools depicting existing processes and systems
- key legal requirements relating to the intellectual property and ownership and privacy of data used, including:
  - Copyright Act 1968 (Cwlth)
  - *Privacy Act 1988* (Cwlth) (APP 1, 3, 6, 8, 9, 11)
  - Privacy and Data Protection Act 2014 (IPP 1, 2, 4, 5, 7, 9, 10)

- file management techniques, including:
  - the use of naming conventions
  - version control
  - backups (full, incremental, differential)
  - security
  - disposal
- ideation techniques and tools for generating design ideas, including:
  - mood boards
  - brainstorming
  - mind maps
  - sketches
  - annotations
- criteria for evaluating design ideas and the efficiency and effectiveness of solutions
- design tools for generating solution designs from design ideas, including:
  - data dictionaries
  - mock-ups
  - object descriptions
  - input-process-output (IPO) charts
  - pseudocode
- characteristics of user experience (UX) and how these affect software design, including:
  - affordance
  - interoperability
  - security (authentication and data protection)
  - usability
- design principles that influence the appearance and functionality of the user interface/s of the software solution, including:
  - alignment
  - balance
  - contrast
  - space
  - text formatting
  - usability
  - navigation.

### Key skills

- document a problem, need or opportunity using a brief
- create, monitor and modify project plans using software
- select and use a range of methods to collect data
- apply analysis tools to determine solution requirements, constraints and scope
- document an analysis as a software requirements specification
- generate design ideas using appropriate ideation techniques and tools
- develop evaluation criteria for design ideas and the efficiency and effectiveness of the software solution
- produce detailed designs using appropriate design principles and tools.

## School-based assessment

### Satisfactory completion

The award of satisfactory completion for a unit is based on whether the student has demonstrated the set of outcomes specified for the unit. Teachers should use a variety of learning activities and assessment tasks to provide a range of opportunities for students to demonstrate the key knowledge and key skills in the outcomes.

The areas of study and key knowledge and key skills listed for the outcomes should be used for course design and the development of learning activities and assessment tasks.

### Assessment of levels of achievement

The student's level of achievement in Unit 3 will be determined by School-assessed Coursework and a School-assessed Task.

School-assessed Coursework tasks must be a part of the regular teaching and learning program and must not unduly add to the workload associated with that program. They must be completed mainly in class and within a limited time frame.

Where teachers provide a range of options for the same School-assessed Coursework task, they should ensure that the options are of comparable scope and demand.

The types and range of forms of School-assessed Coursework for the outcomes are prescribed within the study design. The VCAA publishes [Support materials](#) for this study, which include advice on the design of assessment tasks and the assessment of student work for a level of achievement.

Teachers will provide to the VCAA a numerical score representing an assessment of the student's level of achievement. The score must be based on the teacher's assessment of the performance of each student on the tasks set out in the following table.

### Contribution to final assessment

School-assessed Coursework for Unit 3 will contribute 10 per cent to the study score.

Outcomes	Marks allocated	Assessment tasks
<b>Unit 3</b> <b>Outcome 1</b> Interpret teacher-provided solution requirements and designs and use appropriate features of an object-oriented programming language to develop working software modules.	100	<p>In response to teacher-provided solution requirements and designs, develop four working modules with increasing complexity of programming skills.</p> <ul style="list-style-type: none"> <li>Module 1: Simple calculations using arithmetic, logical and conditional operators</li> <li>Module 2: Reading and writing files</li> <li>Module 3: Sorting and searching with functions or methods</li> <li>Module 4: Classes and objects</li> </ul> <p>At least two modules must include a GUI.</p> <p>All modules must include testing.</p> <p>Task time allocated should be at least 8–14 lessons.</p>
<b>Total marks</b>	<b>100</b>	

## School-assessed Task

The student's level of achievement in Unit 3, Outcome 2 and in Unit 4, Outcome 1 will be assessed through a School-assessed Task. Details of the School-assessed Task for Units 3 and 4 are provided on [page 67](#) of this study design.

## External assessment

The level of achievement for Units 3 and 4 is also assessed by an end-of-year examination (see [page 68](#)), which will contribute 50 per cent to the study score.

## Unit 4: Software development

In this unit, students focus on how the needs of individuals and organisations are met through the development of software solutions using an object-oriented programming (OOP) language and consider the cyber security risks to organisations as a result of insecure software development practices.

In Area of Study 1, students apply the problem-solving stages of development and evaluation to develop their preferred design generated in Unit 3, Area of Study 2 into a working software solution. They test and evaluate the solution and assess the project plan. Unit 4, Area of Study 1 forms the second part of the School-assessed Task (SAT). In Area of Study 2, students examine the current software development practices of an organisation and the risks associated with insecure software development environments and practices. Students evaluate the current security practices and make recommendations to ensure software development environments and practices are secure.

### Software tools

The following table indicates the software tool that students are required to both study and use in this unit.

Area of Study 1	An appropriate OOP language
-----------------	-----------------------------

The following table indicates the software tools that students are required to use, but are not required to study, in this unit.

Area of Study 1	An appropriate tool for documenting and modifying project plans
	Programming tools and/or integrated development environments to facilitate programming and testing of solutions

A prescribed list of software tools and functions, and outcome-specific requirements, will be published annually by the VCAA in the [VCAA Bulletin](#). Schools are to select software tools that fulfil these requirements.

## Area of Study 1

### Software development: development and evaluation

In this area of study, students develop the solution they designed in Unit 3, Area of Study 2 into a software solution that meets their identified problem, need or opportunity by applying the problem-solving stages of development and evaluation. Details of the problem-solving methodology specifications are on [pages 18–23](#).

Appropriate features of an OOP language are used to develop an efficient and effective software solution. Validation, debugging and testing techniques are used to ensure the software solution meets requirements.

Students prepare a beta testing strategy that focuses on the core features of their software solution. The tests must be undertaken by at least two potential users, with the results recorded. Potential users could be other students acting as real users of the solution. Students recommend necessary adjustments to their solution based on the results of their beta tests.

Students continue to monitor and record the progress of their projects using the project plan developed in Unit 3, Area of Study 2. Details could include actual versus expected durations, achievement of milestones

and annotations to explain progress. Students evaluate the quality of their software solution using the evaluation criteria developed in Unit 3, Area of Study 2 and assess the effectiveness of their project plan in developing their project.

Students apply computational thinking skills when developing their design ideas into a software solution.

## Outcome 1

On completion of this unit the student should be able to develop and evaluate a software solution that meets requirements and assess the effectiveness of the project plan.

To achieve this outcome the student will draw on key knowledge and key skills outlined in Area of Study 1.

### Key knowledge

- characteristics of efficient and effective solutions, including:
  - user-centred design
  - clear and concise code
  - detailed internal documentation
- characteristics of data types, data structures and data sources for input, storage and output
- features of a programming language, including:
  - local and global variables and constants
  - data types
  - instructions and control structures (sequence, selection, iteration/repetition)
  - arithmetic, logical and conditional operators
  - graphical user interfaces (GUIs)
  - functions and methods
  - classes and objects
  - access modifiers (public, protected and private)
- established and innovative approaches to software development, including:
  - the use of code repositories
  - application programming interfaces (APIs) and libraries
  - artificial intelligence-based (AI) assistants
- validation techniques, including:
  - existence checking
  - type checking
  - range checking
- debugging and alpha testing techniques for checking that solutions meet requirements and function correctly, including the use of:
  - breakpoints
  - commenting out code
  - relevant test data
  - test cases comparing expected and actual output in testing tables

- strategies for conducting beta testing, including:
  - construction of a testing plan and test scenarios
  - observation of testing scenarios
  - documentation of test results
- features of evaluation strategies, including:
  - evaluation criteria
  - time frame
  - responsibility
- techniques for applying evaluation criteria
- factors that influence the effectiveness of project plans, including:
  - scope creep
  - personnel changes
  - technical issues
- techniques for recording the progress of projects, including:
  - adjustments to tasks
  - adjustments to time frames
  - annotations to project plans
  - monitoring and documenting progress using logs/journals
- techniques for assessing the effectiveness of a project plan, including:
  - reviewing the number of changes made to the project plan during the project
  - the reason changes were necessary
  - the impact of changes on the completion of the project.

### Key skills

- monitor, modify and annotate project plans as necessary
- develop a software solution and write internal documentation
- use and apply appropriate data types, data structures and data sources
- develop and apply suitable naming conventions and validation techniques
- select and apply debugging and alpha testing techniques
- prepare and conduct beta testing using appropriate techniques, capture results and recommend modifications to the software solution to address identified issues
- evaluate the efficiency and effectiveness of the software solution
- assess the effectiveness of the project plan.

## Area of Study 2

### Cyber security: secure software development practices

The secure development of software is crucial within the modern technology and development landscape. Insecure software development environments are problematic for organisations that develop software, regardless of whether solution development occurs internally or is conducted by a secondary organisation. In this area of study, students focus on the security risks to organisations as a result of insecure software development environments and practices. Students analyse and evaluate the security of current software development practices, examine the vulnerabilities and risks using threat-modelling principles and consider

the consequences of identified issues. Students should consider how these risks may be minimised or mitigated before recommending strategies to improve current practices, taking into account the key legal requirements and ethical issues faced by organisations. Frameworks such as the Essential Eight and the Information Security Manual are also taken into account for students to follow current industry practice when considering strategies for improvements.

Students apply systems thinking skills when analysing and evaluating software development security strategies within an organisation, and when recommending strategies to improve current practices.

## Outcome 2

On completion of this unit the student should be able to respond to a teacher-provided case study to analyse an organisation's software development practices, identify and evaluate current security controls and threats to software development practices, and make recommendations to improve practices.

To achieve this outcome the student will draw on key knowledge and key skills outlined in Area of Study 2.

### Key knowledge

- goals and objectives of medium and large organisations
- advantages and disadvantages of developing software in-house or externally
- types of vulnerabilities and risks within insecure development environments, including:
  - use of application programming interfaces (APIs)
  - malware
  - unpatched software
  - poor identity and access management practices
  - man-in-the-middle attacks
  - insider threats
  - cyber security incidents
  - risks present from software acquired by third parties
  - ineffective code review practices
  - combined development, testing and production environments
- security controls used to protect software development practices and data stored within applications, including:
  - version control and code repositories
  - robust identity and access management
  - encryption
  - code review
  - regular updates and patches to software
  - separated development, testing and production environments
- threat modelling principles, including:
  - defining security requirements
  - identifying and mitigating threats
  - confirming threats have been mitigated
- criteria for evaluating the security of software development practices within an organisation
- key legislation and industry frameworks that affect how organisations develop software and control the security and communication of data, including the:
  - Copyright Act 1968 (Cwlth)
  - Essential Eight



- Information Security Manual (ISM) (Guidelines for Software Development: Development, testing and production environments; Secure software design and development; Application security testing)
- *Privacy Act 1988* (Cwlth) (APP 1, 6, 8, 9, 11)
- Privacy and Data Protection Act 2014 (IPP 1, 2, 4, 5, 9)
- ethical issues that arise when developing software, including:
  - ineffective security practices
  - use of artificial intelligence during development
  - intellectual property
  - copyright issues
- mitigation measures to reduce or eliminate threats, vulnerabilities and risks within organisations and development environments
- strategies for improving the security of software development practices, including:
  - onboarding/induction practices and developer training focused on secure development
  - development of risk management plans.

### Key skills

- analyse and describe an organisation's software development practices
- propose and apply criteria to evaluate the effectiveness of the current software development practices
- identify and describe vulnerabilities and risks based on current practices
- identify and discuss the possible legal and ethical consequences to an organisation for ineffective software development practices, and how these could be resolved
- recommend and justify improvements to organisations and their development environments to enhance secure software development practices.

## School-based assessment

### Satisfactory completion

The award of satisfactory completion for a unit is based on whether the student has demonstrated the set of outcomes specified for the unit. Teachers should use a variety of learning activities and assessment tasks to provide a range of opportunities for students to demonstrate the key knowledge and key skills in the outcomes.

The areas of study and key knowledge and key skills listed for the outcomes should be used for course design and the development of learning activities and assessment tasks.

### Assessment of levels of achievement

The student's level of achievement in Unit 4 will be determined by School-assessed Coursework and a School-assessed Task.

School-assessed Coursework tasks must be a part of the regular teaching and learning program and must not unduly add to the workload associated with that program. They must be completed mainly in class and within a limited time frame.

Where teachers provide a range of options for the same School-assessed Coursework task, they should ensure that the options are of comparable scope and demand.

The types and range of forms of School-assessed Coursework for the outcomes are prescribed within the study design. The VCAA publishes [Support materials](#) for this study, which include advice on the design of assessment tasks and the assessment of student work for a level of achievement.

Teachers will provide to the VCAA a numerical score representing an assessment of the student's level of achievement. The score must be based on the teacher's assessment of the performance of each student on the tasks set out in the following table.

## Contribution to final assessment

School-assessed Coursework for Unit 4 will contribute 10 per cent to the study score.

Outcomes	Marks allocated	Assessment tasks
<b>Unit 4</b> <b>Outcome 2</b> Respond to a teacher-provided case study to analyse an organisation's software development practices, identify and evaluate current security controls and threats to software development practices, and make recommendations to improve practices.	100	The student's performance will be assessed using one of the following: <ul style="list-style-type: none"> <li>• structured questions</li> <li>• a report in written format</li> <li>• a report in multimedia format.</li> </ul> The case study scenario needs to enable: <ul style="list-style-type: none"> <li>• an analysis of the organisation's software development practices</li> <li>• an evaluation of the current security controls and threats</li> <li>• recommendations to improve practices.</li> </ul> Task time allocated should be 100–120 minutes.
<b>Total marks</b>	<b>100</b>	

## School-assessed Task

The student's level of achievement in Unit 3, Outcome 2 and in Unit 4, Outcome 1 will be assessed through a School-assessed Task.

The School-assessed Task contributes 30 per cent to the study score.

Outcomes	Assessment tasks
<b>Unit 3</b> <b>Outcome 2</b> Document a problem, need or opportunity, formulate a project plan, document an analysis, and generate design ideas and a preferred design for creating a software solution.	A brief outlining the proposed solution and a project plan (Gantt chart) indicating tasks, times, milestones, dependencies and the critical path <b>AND</b> Analytical tools that depict the interactions between systems, users, data and networks <b>AND</b> An analysis that defines the requirements, constraints and scope of a solution in the form of a software requirements specification <b>AND</b> A folio of design ideas and evaluation criteria <b>AND</b> Detailed design specifications of the preferred design. Time allocated should be at least 8–10 weeks of class time.
<b>Unit 4</b> <b>Outcome 1</b> Develop and evaluate a software solution that meets requirements and assess the effectiveness of the project plan.	A software solution that meets the software requirements specification <b>AND</b> Preparation and conduction of beta testing <b>AND</b> <ul style="list-style-type: none"> <li>an evaluation of the efficiency and effectiveness of the software solution</li> <li>an assessment of the effectiveness of the project plan (Gantt chart) in monitoring project progress</li> </ul> in one of the following: <ul style="list-style-type: none"> <li>a written report</li> <li>an annotated visual plan.</li> </ul> Time allocated should be at least 8 weeks of class time.

## External assessment

The level of achievement for Units 3 and 4 is also assessed by an end-of-year examination.

## End-of-year examination

### Contribution to final assessment

The examination will contribute 50 per cent to the study score.

### Description

The examination will be set by a panel appointed by the VCAA. All the key knowledge and key skills that underpin the outcomes in Units 3 and 4 are examinable.

### Conditions

The examination will be completed under the following conditions:

- Duration: 2 hours
- Date: end-of-year, on a date to be published annually by the VCAA
- VCAA examination rules will apply. Details of these rules are published annually in the [VCE Administrative Handbook](#).
- The examination will be marked by assessors appointed by the VCAA.

### Further advice

The VCAA publishes specifications for all VCE examinations on the VCAA website. Examination specifications include details about the sections of the examination, their weighting, the question format(s) and any other essential information. The specifications are published in the first year of implementation of the revised Unit 3 and 4 sequence together with any sample material.