

Updated September 2019

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

ISBN: 978-1-925676-89-1

© Victorian Curriculum and Assessment Authority 2019

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](https://www.vcaa.vic.edu.au/Footer/Pages/Copyright.aspx#schools). For more information go to: [www.vcaa.vic.edu.au/Footer/Pages/Copyright.aspx](http://www.vcaa.vic.edu.au/Footer/Pages/Copyright.aspx).

The VCAA provides the only official, up-to-date versions of VCAA publications. Details of updates can be found on the VCAA website: [www.vcaa.vic.edu.au](https://www.vcaa.vic.edu.au/Pages/HomePage.aspx).

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: vcaa.copyright@edumail.vic.gov.au.

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.

The VCAA logo is a registered trademark of the Victorian Curriculum and Assessment Authority.

Contents

[Important information 1](#_Toc535224478)

[Introduction 2](#_Toc535224481)

[Scope of study 2](#_Toc535224482)

[Rationale 2](#_Toc535224483)

[Aims 2](#_Toc535224484)

[Structure 3](#_Toc535224485)

[Entry 3](#_Toc535224486)

[Duration 3](#_Toc535224487)

[Changes to the study design 3](#_Toc535224488)

[Monitoring for quality 4](#_Toc535224489)

[Safety and wellbeing 4](#_Toc535224490)

[Use of digital resources 4](#_Toc535224491)

[Employability skills 4](#_Toc535224492)

[Legislative compliance 4](#_Toc535224493)

[Assessment and reporting 5](#_Toc535224494)

[Satisfactory completion 5](#_Toc535224495)

[Levels of achievement 5](#_Toc535224496)

[Authentication 6](#_Toc535224499)

[Key concepts 7](#_Toc535224500)

[Terms used in this study 8](#_Toc535224501)

[Units 1 to 4: Problem-solving methodology 13](#_Toc535224502)

[Units 1 to 4: Problem-solving methodology specifications 14](#_Toc535224503)

[Unit 1: Applied computing 17](#_Toc535224504)

[Area of Study 1 17](#_Toc535224505)

[Area of Study 2 19](#_Toc535224507)

[Assessment 20](#_Toc535224510)

[Unit 2: Applied computing 21](#_Toc535224511)

[Area of Study 1 21](#_Toc535224512)

[Area of Study 2 23](#_Toc535224515)

[Assessment 24](#_Toc535224518)

[Unit 3: Data analytics 26](#_Toc535224519)

[Area of Study 1 26](#_Toc535224521)

[Area of Study 2 28](#_Toc535224523)

[School-based assessment 29](#_Toc535224526)

[School-assessed Task 30](#_Toc535224528)

[External assessment 30](#_Toc535224529)

[Unit 4: Data analytics 31](#_Toc535224530)

[Area of Study 1 31](#_Toc535224532)

[Area of Study 2 32](#_Toc535224535)

[School-based assessment 34](#_Toc535224537)

[School-assessed Task 35](#_Toc535224539)

[External assessment 36](#_Toc535224540)

[Unit 3: Software development 37](#_Toc535224542)

[Area of Study 1 37](#_Toc535224543)

[Area of Study 2 39](#_Toc535224545)

[School-based assessment 40](#_Toc535224548)

[School-assessed Task 41](#_Toc535224550)

[External assessment 41](#_Toc535224551)

[Unit 4: Software development 42](#_Toc535224552)

[Area of Study 1 42](#_Toc535224553)

[Area of Study 2 44](#_Toc535224555)

[School-based assessment 45](#_Toc535224558)

[School-assessed Task 47](#_Toc535224560)

[External assessment 47](#_Toc535224561)

Important information

Accreditation period

Units 1–4: 1 January 2020 – 31 December 2024

Implementation of this study commences in 2020.

Other sources of information

The [*VCAA Bulletin*](https://www.vcaa.vic.edu.au/news-and-events/bulletins-and-updates/bulletin/Pages/index.aspx) 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’s website at: [www.vcaa.vic.edu.au](https://www.vcaa.vic.edu.au/Pages/HomePage.aspx).

To assist teachers in developing courses, the VCAA publishes online the Advice for teachers, which includes teaching and learning activities for Units 1 to 4, and advice on assessment tasks and performance level descriptors for School-assessed Coursework
in Units 3 and 4.

The current [*VCE and VCAL Administrative Handbook*](https://www.vcaa.vic.edu.au/administration/vce-vcal-handbook/Pages/index.aspx) 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

VCE schools may reproduce parts of this study design for use by teachers. The full VCAA Copyright Policy is available at: [www.vcaa.vic.edu.au/Footer/Pages/Copyright.aspx](http://www.vcaa.vic.edu.au/Footer/Pages/Copyright.aspx).

Introduction

Scope of study

VCE Applied Computing focuses on the strategies and techniques for creating digital solutions to meet specific needs and to manage the threats to data, information and software security. The study examines the attributes of each component of an information system including people, processes, data and digital systems (hardware, software, networks), and how their interrelationships affect the types and quality of digital solutions.

VCE Applied Computing is underpinned by four key concepts: digital systems, data and information, approaches to problem solving, and interactions and impact.

VCE Applied Computing provides students with opportunities to acquire and apply knowledge and skills to use digital systems efficiently, effectively and innovatively when creating digital solutions. Students investigate legal requirements and ethical responsibilities that individuals and organisations have with respect to the security and integrity of data and information. Through a structured approach to problem solving, incorporating computational, design and systems thinking, students develop an awareness of the technical, social and economic impacts of information systems, both currently and into the future.

Rationale

Technology continues to evolve rapidly, providing opportunities for enterprising individuals to create new technologies and innovative uses for existing technologies. This study equips students with the knowledge and skills required to adapt to a dynamic technological landscape, including the ability to identify emerging technologies, envisage new uses for digital technologies and consider the benefits that these technologies can bring to society at a local and at a global level.

VCE Applied Computing facilitates student-centred learning that enables students to build capabilities in critical and creative thinking, and to develop communication and collaboration, and personal, social and information and communications technology (ICT) 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 provides a pathway to further studies in areas such as business analysis, computer science, cybersecurity, data analytics and data science, data management, games development, ICT, networks, robotics, software engineering and telecommunications, and other careers relating to digital technologies.

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 cybersecurity, data analytics and programming
* apply the problem-solving methodology to analyse needs and opportunities, design and develop solutions to problems and evaluate how effectively solutions meet needs and opportunities
* apply project management techniques to assist with the development of digital solutions
* develop an informed perspective on current and emerging digital technologies and disseminate findings
* identify and evaluate innovative and emerging opportunities for digital solutions and technologies
* develop critical and creative thinking, communication and collaboration, and personal, social and ICT skills.

Structure

The study is made up of six units.

Unit 1: Applied computing

Unit 2: Applied computing

Unit 3: Data analytics

Unit 4: Data analytics

Unit 3: Software development

Unit 4: Software development

Note: students may elect to undertake one or both of these Units 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 to 4 are designed to a standard equivalent to the final two years of secondary education. All VCE studies are benchmarked against comparable national and international curriculum.

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*](https://www.vcaa.vic.edu.au/news-and-events/bulletins-and-updates/bulletin/Pages/index.aspx). 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 and VCAL Administrative Handbook*](https://www.vcaa.vic.edu.au/administration/vce-vcal-handbook/Pages/index.aspx). 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.

Use of digital resources

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

* appropriate laptop, notebook or desktop computers
* printers
* internet
* a range of software tools including:
* database management software to format, store, edit and retrieve data
* spreadsheet and/or statistical software to format, store, edit and retrieve data
* software to edit, retrieve, insert and delete data such as text, sound, static and moving images to create multimodal presentation of information
* drawing and graphic software and data visualisation software to create graphical output
* tools to provide programming environments.

Employability skills

This study offers a number of opportunities for students to develop employability skills. The *Advice for teachers* support material provides 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* and *Copyright Act 1968*, must be met.

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 and receive 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 took the study. Teachers should refer to the current [*VCE and VCAL Administrative Handbook*](https://www.vcaa.vic.edu.au/administration/vce-vcal-handbook/Pages/index.aspx) 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 and VCAL Administrative Handbook*](https://www.vcaa.vic.edu.au/administration/vce-vcal-handbook/Pages/index.aspx) for authentication procedures.

Key concepts

Four key concepts underpin VCE Applied Computing: digital systems, data and information, approaches to problem solving, and interactions and impact. These concepts are used as a way to understand and structure the Applied Computing content. Within an area of study, the key knowledge is organised under the headings of relevant concepts; however, not all concepts may be addressed in any one area of study.

**Digital systems** focuses on the functions and technical underpinnings of hardware and software components, as well as networks and the internet, including protocols. This concept focuses on how hardware and software are used to manage and control access to secure data. Digital systems form one of the components of an information system, along with people, data and processes.

**Data and information** focuses on the nature of data and how it can be acquired, structured, represented and interpreted to extract meaning to produce information. This process of preparing data and information appropriately is the precursor to creating solutions that meet the intended needs of individuals and organisations.

**Approaches to problem solving** focuses on the creation of solutions and the presentation of findings in response to a problem, need or opportunity. A detailed problem-solving methodology is on [pages 12 to 15](#ProblemMethodology). Approaches to problem solving include computational thinking, design thinking and systems thinking (see Terms used in this study on [pages
8 to 11](#Terms)).

**Interactions and impact** focuses on the relationships within and between information systems and the effectiveness of these in achieving organisational goals and objectives. Relationships are considered from three perspectives: how people interact with other people when using digital systems, including for communication and collaboration; how people interact with, or respond to, different types of digital systems; and how information systems interact with other information systems. This concept further considers the impact of these relationships on meeting current and future needs of individuals, organisations and society, including the ownership and privacy of data and information, and personal safety.

Terms used in this study

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

|  |  |
| --- | --- |
| **Term** | **Definition** |
| Computational thinking | The process of recognising aspects of computation in the world and being able to think logically, algorithmically, recursively and inferentially. It typically involves inferential thinking, defining problems through decomposition, documenting steps and decisions through algorithms, the use of programming languages and software, and evaluating the resulting solutions. |
| Conventions | General or formal agreements to ensure consistency and clarity when working with digital systems. Program names and file names are examples of conventions to be used during software development. Conventions can be mandatory, such as when passwords must have specific characters, and recommended, such as including a subject line in emails. |
| Cybersecurity | An interdisciplinary field, which includes aspects of ethics, government, law, organisational policies and risk management. It consists of defensive methods used to detect and stop would-be intruders and to enable the unimpeded operation of digital systems and technologies. |
| Data analytics | The processes and tools that allow organisations to acquire and/or extract data in various forms, analyse the data to identify patterns or trends, draw inferences about the data and present findings using visual methods that provide clear and unambiguous conclusions.  |
| Data structures | The way data is stored to enable efficient algorithms to be used to optimise program execution time and memory usage. Types of data structures include: arrays, associative arrays, classes, fields, files, hash tables, linked lists, queues, records and stacks. |
| Data types | The forms that an item of data can take, including binary (as represented in images and sound), Boolean, character and numeric, characterised by the kind of operations that can be performed on it. Depending on the software being used, these data types can be divided into more specific data types, for example integer and floating point, which are numeric types. More sophisticated types can be derived from them, for example a string of characters or a data type, and their names may vary, such as text data type versus string data type. |
| Data visualisations | The result of a process of using software tools to select and access data from large repositories to present the data as a graphic representation usually in the form of charts, histograms, graphs, maps, network diagrams and spatial relationships diagrams. Data visualisations help 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. |

|  |  |
| --- | --- |
| Design principles | Design principles are accepted characteristics that contribute to the functionality, usability and appearance of solutions. In this study the principles related to usability include ease of use, flexibility and robustness, and accessibility, including navigation and error tolerance. Design principles related to appearance are alignment, balance, contrast, image use, space, and text and table formatting. |
| 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, and 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.  |
| Effectiveness | A measure of how well a solution, information management strategy or a network functions and whether each achieves its intended results. Measures of effectiveness in a solution include accessibility, accuracy, attractiveness, clarity, communication of message, completeness, readability, relevance, timeliness, and usability. Measures of effectiveness of an information management strategy include currency of files, ease of retrieval, integrity of data and security. Measures of effective networks include maintainability, reliability and the security of data during storage and transmission. |
| 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 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. |
| 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. |
| Infographics | Graphical representations of complex data or information. They rely upon visual elements to quickly and clearly communicate patterns or trends in data or information. These include complementary colour schemes, easy-to-read fonts, graphs, simple charts and statistics. |
| Information system | The combination of digital hardware, software and network components (digital systems), data, processes and people that interact to create, control and communicate ideas and data in digital solutions. |
| Internet of Things (IoT) | The network formed by ‘smart devices’ such as mobile phones, wearable devices, headphones, and other devices that communicate and interact over the internet and/or to other IoT devices. These devices must have a method of connecting to the internet, whether by a wireless or wired technology for the purposes of data transmission. |

|  |  |
| --- | --- |
| Legal requirements | Individuals and organisations are expected to comply with legal requirements, with respect to the ownership and privacy of information, and freedom of expression. For the purposes of this study, the key provisions of the following acts are relevant: *Charter of Human Rights and Responsibilities Act 2006* (Vic) (sections 13, 14 and 15), *Copyright Act 1968*, *Health Records Act 2001*, *Privacy Act 1988*, including *Privacy Amendment (Enhancing Privacy Protection) Act 2012* and *Privacy Amendment (Notifiable Data Breaches) Act 2017*, *Privacy and Data Protection Act 2014*, and the *Spam Act 2003* (Part 1.3, Simplified outline). |
| Managing files | File management includes the naming, storing and handling of files. Strategies to manage files should have a consistent and descriptive naming convention and a clear directory structure. Files should be easily identified and accessed. File management also includes procedures for backing up, archiving and deleting files. |
| Naming conventions | A set of guidelines for providing consistency in the naming of entities in software development and data management. These include program names, function names, variable names, table names, file names, and document names. Naming conventions can minimise the effort needed to read and understand source code.  |
| Physical security controls | The equipment and procedures used to assist in the protection of information systems and the files created, communicated and stored by individuals and organisations. Equipment controls include zoned security strategies, barrier techniques and biometrics. Physical procedures include backing up, shredding confidential documents and checking authorisation credentials. Also see Software security controls. |
| Processing features | The distinctive or characteristic parts of a programming language. These include classes, control structures, functions, instructions and methods.  |
| Project management | The practice of applying techniques, processes, tools, knowledge and skills to deliver a solution. Features of project management include identification of tasks, sequencing, time allocation, dependencies, milestones and critical path. |
| Pseudocode | 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. These include the use of START/BEGIN and FINISH/END to indicate the beginning and end of a program, IF…ELSE and SELECT to represent decisions, and REPEAT…UNTIL/WHILE…DO for loops. |
| Security threats | The actions, devices and events that threaten the integrity and security of data and information stored within, and communicated between, information systems. The threats can be accidental, such as losing a portable storage device containing files; deliberate, such as malware, phishing; and events-based such as a power surge. |
| 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, system and technical requirements, constraints, scope and assumptions. |
| Software security controls | The software and procedures used to assist in the protection of information systems and the files created, communicated and stored by individuals and organisations. These include usernames and passwords, access logs and audit trails, access restrictions, encryption, firewalls and system protection, and security protocols such as Transport Layer Security (TLS). |

|  |  |
| --- | --- |
| Solution (digital) | The method of creating required digital output through the application of digital systems and processes that transforms data into information. The output of a solution may take the form of an information product, such as a website, instructions to control a game, an abstract piece of art or a soundscape. Solutions can be interactive or non-interactive, online (internet connected) or not, multimodal or not.  |
| 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. This approach enables students to understand whole systems and work with complexity, uncertainty and risk. |
| Testing techniques | Techniques to ensure that a solution is operating as intended. Test data should include data types and values that rigorously test software functions. Testing tables can be used to record the outcomes of tests. |
| Types of data | Types of data are general categories of data, including image (still and moving), number and text. |
| Unified modelling language (UML) | A visual modelling language that describes the interfaces between solutions, users and networks. An example of this is a use case diagram (UCD) that models the requirements of an information system. |
| Validation | The checking of data for its reasonableness and completeness. Validation of data includes: existence or presence checking, which verifies that a required field has a value entered and is not empty or blank; range checking, which involves ensuring that data entered falls within a certain range; and type checking, which confirms that data entered is of a particular type. |
| Verification | The checking of data after data entry. This can include proofreading to ensure that the data entered matches the source data. |

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 are provided in the introduction to relevant areas of study. Note: when creating solutions, this methodology can be applied as a single stage-by-stage problem-solving process or can be applied to each iteration of an agile problem-solving process.



Figure 1: A problem-solving methodology for creating digital solutions

Units 1 to 4: Problem-solving methodology specifications

|  |  |  |
| --- | --- | --- |
| **Stage/Activities** | **Description** | **Unit application** |
| **1**  | **2**  | **3** | **4**  |
| **Analysis:** The Analysis Stage involves determining what is required to solve a problem. It involves identifying the solution requirements, constraints and scope. |
|  | Solution requirements | Solution requirements can be described as being 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. These include: 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 taken into account when designing a solution.  |
| Determine the constraints of the solution. These include economic, such as cost and time; technical, such as speed of processing, capacity, availability of equipment, compatibility and security; social, such as level of expertise of users; legal, such as ownership and privacy of data requirements; and usability, such as usefulness and ease of use of solutions. | **•** | **•** | **•** |  |
| 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 | Identify and clarify the data and information that needs to be collected and from what sources it will be collected. |  | **•** | **•** |  |
| Critically analyse the sources of data and information to determine the reliability of it. |  | **•** | **•** |  |
| Draft and evaluate questions to critically analyse requirements, needs or opportunities. |  | **•** | **•** |  |
| Develop strategies for asking follow-up questions to further clarify the data and information collected. |  | **•** | **•** |  |

|  |  |  |
| --- | --- | --- |
| **Stage/Activities** | **Description** | **Unit application** |
| **1**  | **2**  | **3** | **4**  |
| **Design:** The Design Stage involves determining how the solution requirements will function and appear. It involves developing evaluation criteria; designing the functionality, appearance and user interface of the solution; and designing the tests to ensure that requirements can be met. |
|  | Solution design  | 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 design tools, including data dictionaries, input–process–output (IPO) charts, object descriptions and pseudocode.  | **•** | **•** | **•** |  |
| Use design tools to show the relationships between the components of a solution, including storyboards, site maps, structure charts, context diagrams and data flow diagrams. |  | **•** | **•** |  |
| Design the visual components of the solution, including the user interface, reports and any graphic representations or data visualisations generated.  | **•** | **•** | **•** |  |
| Identify the position and size of text, images and graphics, font types and styles, colours and text enhancements.  | **•** | **•** | **•** |  |
| Use design tools, including layout diagrams, annotated diagrams and mock-ups. | **•** | **•** | **•** |  |
| Design tests to ensure the solution will achieve what it is intended to do.  | **•** | **•** | **•** |  |
| Determine what tests need to be performed, what test data needs to be included and the expected result after the test is conducted.  | **•** | **•** | **•** |  |
| *Note that testing of the actual solution as it is developed is performed during the Development Stage*. |
| Skills underpinning the Solution design activity | Investigate possible solutions and develop connections between ideas. |  |  | **•** |  |
| Use a range of techniques to generate a range of alternative solutions designs. |  |  | **•** |  |
| Create and connect initial design ideas using text and diagrams. |  | **•** | **•** |  |
| Determine possible contingencies when developing solution designs to mitigate issues. |  | **•** | **•** |  |
| Evaluation criteria | Develop evaluation criteria to determine the degree to which solution designs meet requirements. Evaluation criteria measure the efficiency and effectiveness of the designs.  |  | **•** | **•** |  |
| *Note that the same evaluation criteria are used in the Evaluation Stage to determine the degree to which the solution meets requirements. The evaluation criteria, in the Evaluation Stage, measure the efficiency and effectiveness of the solution*. |
| Modify and refine the evaluation criteria as the solution develops. |  | **•** | **•** |  |

|  |  |  |
| --- | --- | --- |
| **Stage/Activities** | **Description** | **Unit application** |
| **1**  | **2**  | **3** | **4**  |
| **Development:** The Development Stage 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 | Develop the solution through the process of manipulation.  | **•** | **•** |  | **•** |
| Use appropriate software functions and techniques, formats and conventions, suitable algorithms and processing features. | **•** | **•** |  | **•** |
| Validation | Use validation techniques to check data entry for reasonableness and completeness of data, including existence or presence check, range check and type check. |  | **•** |  | **•** |
| Use verification techniques after data entry to ensure that data entered matches the source data, including proofreading.  |  | **•** |  | **•** |
| *Note that the effectiveness of validation is determined through the testing activity below*. |
| Testing | Develop a testing strategy to ensure that the solution works as intended.  | **•** | **•** |  | **•** |
| Compare the actual results of the tests against the expected results from the Design Stage.  | **•** | **•** |  | **•** |
| Conduct tests, record the actual results and correct any identified errors. | **•** | **•** |  | **•** |
| Documentation | Write internal documentation to support the functioning, maintenance and upgrading of the solution and to support the use of the solution by end-users. | **•** | **•** |  | **•** |
| Skills underpinning the Design Stage  | Determine possible contingencies when developing solutions to mitigate issues. |  | **•** |  | **•** |
| Modify and evolve design ideas as the solution develops. |  | **•** |  | **•** |
| Modify and refine the evaluation criteria as the solution develops. |  | **•** |  | **•** |
| **Evaluation:** The Evaluation Stage involves determining the degree to which the software solution has met requirements. It involves evaluating the software solution against the evaluation criteria to see how well it meets requirements and a strategy to evaluate the software solution after implementation. |
|  | Solution evaluation | 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 Solution evaluation activity | Critically evaluate the analysis process and discuss how it assisted in meeting the requirements, 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 process differently. |  | **•** |  | **•** |
| Evaluation strategy  | Proposal of a strategy to evaluate the extent to which the solution meets the needs of the user after the solution has been developed and implemented.  |  |  |  | **•** |
| Specify a timeline and outline the data to be collected and by what methods and techniques, and how the data relates to the evaluation criteria. |  |  |  | **•** |

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 programming languages to develop working software solutions.

In Area of Study 1, as an introduction to data analytics, students respond to a teacher-provided analysis of requirements and designs to identify and collect data in order to present their findings as data visualisations. They present work that includes database, spreadsheet and data visualisations solutions. In Area of Study 2 students select and use a programming language to create a working software solution. Students prepare, document and monitor project plans and engage in all stages of the problem-solving methodology.

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. |
| Area of Study 2 | An appropriate programming language. |

Area of Study 1

Data analysis

In this area of study students use software tools to create data visualisations in response to teacher-provided requirements and designs. The software tools are used for the collection, 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 and spatial relationships diagrams. No restrictions are placed on the software tools used to create data visualisations.

Students examine the features of different design tools to represent the functionality and appearance of software solutions. They interpret given designs and create database, spreadsheet and data visualisations solutions using the data collected. 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 create data visualisations.

**Outcome 1**

On completion of this unit the student should be able to interpret teacher-provided solution requirements and designs, collect and manipulate data, analyse patterns and relationships, 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

*Data and information*

* types and purposes of qualitative and quantitative data
* characteristics of data and information
* sources, methods and techniques for acquiring and referencing primary and secondary data and information
* interpretation of information for communication and decision making
* factors affecting the quality of data and information, such as accuracy, bias, integrity, relevance and reliability
* characteristics of data types and data structures relevant to selected software tools
* procedures for the legal and ethical collection and use of data and information, such as using consent forms
* techniques for protecting data and information from misuse, such as de-identifying personal data and the use of physical and software security controls

*Approaches to problem solving*

* structural characteristics of spreadsheets and databases, such as cells, fields, records and tables
* types and purposes of data visualisations suitable for educating, entertaining, informing and persuading audiences
* functional and non-functional requirements of solutions, constraints and scope
* design tools for representing the functionality and appearance of databases, spreadsheets and data visualisations, such as annotated diagrams and mock-ups
* formats and conventions suitable for databases, spreadsheets and data visualisations
* software functions and techniques for efficiently and effectively manipulating, validating and testing data to develop databases, spreadsheets and data visualisations

*Interactions and impacts*

* Australian Privacy Principles relating to the acquisition, management and communication of data and information including non-identification of individuals (Principle 2), information only being held for its primary purpose (Principle 6) and the security measures used to protect personal information (Principle 11)
* ethical issues arising from the acquisition, storage and use of data and information.

Key skills

* acquire and reference data and information from primary and secondary sources, taking into account legal and ethical considerations
* analyse the selected data, and discuss the relationships and patterns identified
* interpret solution requirements, constraints and scope
* interpret designs using appropriate design tools to represent the functionality and appearance of databases, spreadsheets and data visualisations
* use software, and select and apply functions, formats, conventions, data validation and testing techniques to efficiently manipulate data and create data visualisations
* compare and interpret data visualisations.

Area of Study 2

Programming

In this area of study students use a programming language to create a working software solution in response to teacher-provided solution requirements. Students apply the problem-solving stages of design, development and evaluation to develop the solution. Details of the relevant problem-solving methodology specifications are on [pages 12–15](#ProblemMethodology).

Students apply methods and techniques for creating a working software solution using a range of processing features and data structures. They apply testing and debugging techniques to ensure the software solution works as intended.

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, dependencies, milestones and critical path. They record and monitor the progress of their working solution throughout the stages of the problem-solving methodology. Students do not have to use dedicated project management software.

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, develop and evaluate a software solution using a programming language.

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

Key knowledge

*Digital systems*

* functions and capabilities of key hardware and software components of digital systems required for processing, storing and communicating data and information

*Data and information*

* characteristics of data types
* types of data structures

*Approaches to problem solving*

* features of functional and non-functional solution requirements, constraints and scope
* design tools for representing the functionality and appearance of solution designs such as data dictionaries, mock-ups and pseudocode
* naming conventions for solution elements such as files, functions, methods and variables
* processing features of a programming language
* characteristics of internal documentation
* formatting and structural characteristics of input and output such as file formats
* testing and debugging techniques to ensure software solutions meet requirements such as test tables and test data
* techniques for evaluating the efficiency and effectiveness of software solutions
* project plans to coordinate and monitor the tasks, including sequencing and time allocation to create software solutions.

Key skills

* analyse 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 processing features of a programming language
* design and apply suitable testing and debugging techniques using appropriate test data
* evaluate the efficiency and effectiveness of the software solution to meet requirements
* document and monitor project plans using software.

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 timeframe.

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:

* a folio of exercises or software solutions and a written report
* a presentation (oral, multimedia, visual) to present findings or software solutions.

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

Unit 2: Applied computing

In this unit students focus on developing innovative solutions to needs or opportunities that they have identified, and propose strategies for reducing security risks to data and information in a networked environment.

In Area of Study 1 students work collaboratively and select a topic for further study to create an innovative solution in an area of interest. 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. In Area of Study 2, as an introduction to cybersecurity, students investigate networks and the threats, vulnerabilities and risks to data and information. They propose strategies to protect the data accessed using 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 create an innovative solution, for example a programming language, spreadsheet software, web-authoring software, presentation software, tool for planning a project. |
| Area of Study 2 | A software tool to represent a network. |

Area of Study 1

Innovative solutions

In this area of study students work collaboratively to develop an innovative solution to an identified need or opportunity. They apply all stages of the problem-solving methodology to investigate the use of digital devices and emerging technologies and their applications. Details of the problem-solving methodology specifications are on [pages 12–15](#ProblemMethodology).

The innovative solution may take the form of a proof of concept, prototype or product. Students choose one of the following topics to explore in greater detail:

* artificial intelligence, machine learning or neural networks
* assistive and wearable technologies or Internet of Things (IoT)
* creating with digital systems such as drones, microcontrollers, nanosatellites and robotic devices
* games development, multimedia programming or web authoring
* mixed realities such as augmented and virtual reality
* investigation/research project on innovative uses for emerging technologies such as blockchain
* 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, analyse, design, develop and evaluate an innovative solution to an identified need or opportunity involving a digital system.

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

Key knowledge

*Digital systems*

* components of digital systems
* types of digital devices used for a range of current and emerging applications such as smart phones, smart refrigerators and virtual assistants
* emerging trends in digital systems and the importance of innovation to organisations, such as improving efficiency and effectiveness of customer service and maintaining competitiveness
* functions and capabilities of digital systems used by individuals and organisations, such as assistive technologies, financial services, global positioning system (GPS) devices, robotics and traffic management

*Data and information*

* techniques for collecting data to determine user needs and requirements, such as interviews and surveys

*Approaches to problem solving*

* techniques for documenting the development of solutions
* solution specifications such as functional and non-functional requirements, constraints and scope
* characteristics of creative and innovative solutions
* design tools and techniques for representing solution designs, such as mock-ups, pseudocode, sitemaps and storyboards
* functions and techniques for developing innovative solutions
* techniques for validating and testing solutions
* evaluation criteria and techniques for evaluating the efficiency and effectiveness of innovative solutions
* tools and techniques for coordinating and monitoring projects, such as Gantt charts

*Interactions and impact*

* goals and objectives of digital systems
* economic issues involving emerging technologies, such as access, deskilling, job loss, misuse and sustainability
* the impact of current and emerging technologies, such as automation, cyberbullying and the decline of physical human interactions and interpersonal skills
* key legislation and how emerging technologies are affected by: the *Copyright Act 1968*,the *Health Records Act 2001*,the *Privacy Act 1988* and the *Privacy and Protection Act 2014*
* ethical issues arising from the development of emerging technologies.

Key skills

* investigate a problem, need or opportunity and identify potential users and purpose
* propose 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 the innovative solution
* design and apply suitable validation and 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 need or opportunity
* document, monitor and modify project plans using a Gantt chart.

Area of Study 2

Network security

In this area of study students investigate how networks enable data and information to be exchanged locally and globally. Students examine the hardware and software components and procedures required to connect and maintain wired, wireless and mobile communications technology. They apply this knowledge to design a Local Area Network (LAN), describe its components and explain the transmission of data and information in this network. Students develop an understanding of cybersecurity issues when they investigate the threats, vulnerabilities and risks to data and information stored within and transmitted across networks, and propose strategies for reducing security risks.

Students apply systems thinking skills when designing LANs 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 the capabilities and vulnerabilities of a network, design a network solution, discuss the threats to data and information, and propose strategies to protect the security of data and information.

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

Key knowledge

*Digital systems*

* applications and capabilities of LANs, Wide Area Networks (WANs) and Wireless Personal Area Networks (WPANs)
* functions and characteristics of key hardware and software components of networks required for communicating and storing data and information
* strengths and limitations of wired, wireless and mobile communications technology, measured in terms of cost, data storage options, data transfer rate, reliability and security
* technical underpinnings of intranets, the internet and virtual private networks
* design tools for representing the appearance of networks
* security threats to data and information, such as improper credential management, malicious software, outdated versions of software and weak passwords
* technical underpinnings of malware that can intentionally threaten the security of networks, such as denial of service attacks on websites, spyware, viruses and worms
* data and network protection strategies, such as authentication techniques and symmetric and asymmetric encryption methods
* preventative practices to reduce risks to networks, such as application of firmware, disaster recovery plans, operating system updates, software malware updates and staff procedures
* technical underpinnings of intrusion detection systems (IDS) and intrusion prevention systems (IPS)
* the role of ethical hacking

*Interactions and impacts*

* risks and benefits of using networks in a global environment
* key legislation that affects how organisations control the storage and communication of data and information: the *Health Records Act 2001*, the *Privacy Act 1988* and the *Privacy and Data Protection Act 2014*
* ethical issues arising from data and information security practices.

Key skills

* identify and describe the applications and capabilities of different networks
* examine the impact of common network vulnerabilities
* design a network solution with wireless capability
* identify and evaluate threats to the security of data and information
* propose and justify strategies to protect the security of data and information within a network
* identify and discuss possible legal and ethical issues arising from ineffective data and information security practices.

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 timeframe.

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:

* a presentation (oral, multimedia, visual) of an innovative solution
* a written report
* an annotated visual report
* a case study with structured questions
* the design of a wireless network or a working model of a wireless network.

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

Unit 3: Data analytics

In this unit students apply the problem-solving methodology to identify and extract data through the use of software tools such as database, spreadsheet and data visualisation software to create data visualisations or infographics. 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. Students develop data visualisations and use appropriate software tools to present findings. Appropriate software tools include database, spreadsheet and data visualisation software.
In Area of Study 2 students propose a research question, prepare a project plan, collect and analyse data, and design infographics 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. |
| Area of Study 2 | At least one data manipulation tool and one visualisation tool, for example database software, spreadsheet software, data visualisation software, tool for planning a project. |

A list of minimum software capabilities for database software, spreadsheet software and data visualisation software requirements will be published annually by the VCAA in the [*VCAA Bulletin*](https://www.vcaa.vic.edu.au/Pages/correspondence/index.aspx#bulletin).

Area of Study 1

Data analytics

In this area of study students access, 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 develop software solutions using database, spreadsheet and data visualisation software tools to undertake the problem-solving activities in the development stages of manipulation, validation and testing.

The software solutions involve importing data from files to a database to identify patterns and relationships. Data is then imported into a spreadsheet for further refinement before presenting findings as data visualisations. This will prepare students for creating infographics or dynamic data visualisations in Unit 4, Area of Study 1. 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 respond to teacher-provided solution requirements and designs to extract data from large repositories, manipulate and cleanse data and apply a range of functions to develop software solutions 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

*Data and information*

* techniques for efficient and effective data collection, including methods to collect census, Geographic Information System (GIS) data, sensor, social media and weather
* factors influencing the integrity of data, including accuracy, authenticity, correctness, reasonableness, relevance and timeliness
* sources of, and methods and techniques for, acquiring authentic data stored in large repositories
* methods for referencing primary and secondary sources, including American Psychological Association (APA) referencing system
* characteristics of data types

*Approaches to problem solving*

* methods for documenting a problem, need or opportunity
* methods for determining solution requirements, constraints and scope
* naming conventions to support efficient use of databases, spreadsheets and data visualisations
* a methodology for creating a database structure: identifying entities, defining tables and fields to represent entities; defining relationships by identifying primary key fields and foreign key fields; defining data types and field sizes, normalisation to third normal form
* design tools for representing databases, spreadsheets and data visualisations, including data dictionaries, tables, charts, input forms, queries and reports
* design principles that influence the functionality and appearance of databases, spreadsheets and data visualisations
* functions and techniques to retrieve required information through querying data sets, including searching, sorting and filtering to identify relationships and patterns
* software functions, techniques and procedures to efficiently and effectively validate, manipulate and cleanse data including files, and applying formats and conventions
* types and purposes of data visualisations
* formats and conventions applied to data visualisations to improve their effectiveness for intended users, including clarity of message
* methods and techniques for testing databases, spreadsheets and data visualisations

*Interactions and impact*

* reasons why organisations acquire data.

**Key skills**

* interpret solution requirements and designs to develop data visualisations
* identify, select and extract relevant data from large repositories
* use a standard referencing system to acknowledge intellectual property
* organise, manipulate and cleanse data using database and spreadsheet software
* select, justify and apply functions, formats and conventions to create effective data visualisations
* develop and apply suitable validation and testing techniques to software tools used.

Area of Study 2

Data analytics: analysis and design

In this area of study students, individually, determine and propose a research question and collect and analyse data. This is the first part of the School-assessed Task (SAT), involving analysis and design, with the second part undertaken in Unit 4, Area of Study 1.

Students prepare a project plan, taking into account all stages of the problem-solving methodology covered in this outcome and in Unit 4, Area of Study 1. Details of the relevant problem-solving methodology specifications are on [pages 12–15](#ProblemMethodology). Students do not have to use dedicated project-management software. They determine the milestones of their project plans.

A range of methods is used to collect data for analysis. The analysis is stated in terms of requirements, data to support the research question, constraints and scope. Students generate design ideas for creating their database and/or spreadsheet solutions and infographics or dynamic data visualisations to present findings, which could include annotations to indicate key functions and layouts. Evaluation criteria are developed and applied to select the preferred design idea. These evaluation criteria will be used in Unit 4, Area of Study 1, to evaluate the efficiency and effectiveness of the infographics or dynamic data visualisations.

Students apply computational thinking skills when analysing the data associated with a research question and apply design thinking skills when designing infographics 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 analyse data, generate alternative design ideas and represent the preferred design for creating infographics 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**

*Digital systems*

* roles, functions and characteristics of digital system components
* physical and software security controls used by organisations for protecting stored and communicated data

*Data and information*

* primary and secondary data sources and methods of collecting data, including interviews, observation, querying of data stored in large repositories and surveys
* techniques for searching, browsing and downloading data sets
* suitability of quantitative and qualitative data for manipulation
* characteristics of data types and data structures relevant to selected software tools
* methods for referencing secondary sources, including the APA referencing system
* criteria to check the integrity of data, including accuracy, authenticity, correctness, reasonableness, relevance and timeliness
* techniques for coding qualitative data to support manipulation

*Approaches to problem solving*

* features of a research question, including a statement identifying the research question as an information problem
* functional and non-functional requirements, including data to support the research question, constraints and scope
* types and purposes of infographics and dynamic data visualisations
* design principles that influence the appearance of infographics and the functionality and appearance of dynamic data visualisations
* design tools for representing the appearance and functionality of infographics and dynamic data visualisations, including data manipulation and validation, where appropriate
* techniques for generating alternative design ideas
* criteria for evaluating alternative design ideas and the efficiency and effectiveness of infographics or dynamic data visualisations
* features of project management using Gantt charts, including the identification and sequencing of tasks, time allocation, dependencies, milestones and the critical path

*Interactions and impact*

* key legal requirements for the storage and communication of data and information, including human rights requirements, intellectual property and privacy.

Key skills

* frame a research question
* analyse and document solution requirements, constraints and scope of infographics or dynamic data visualisations
* apply techniques for searching, downloading, browsing and referencing data sets
* select and apply design tools to represent the functionality and appearance of infographics or dynamic data visualisations
* generate alternative design ideas
* develop evaluation criteria to select and justify preferred designs
* produce detailed designs using appropriate design methods and techniques
* propose and apply appropriate methods to secure stored data
* create, monitor and modify project plans using software.

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 timeframe.

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 *Advice for teachers* for this study, which includes 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** |
| **Unit 3****Outcome 1**Respond to teacher-provided solution requirements and designs to extract data from large repositories, manipulate and cleanse data and apply a range of functions to develop software solutions to present findings. | **100****100** | In response to teacher-provided solution requirements and designs, create software solutions. |
| **Total marks** | **100** |  |

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 34](#SAT) of this study design.

External assessment

The level of achievement for Units 3 and 4 is also assessed by an end-of-year examination, 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 or dynamic data visualisations based on large complex data sets and on the security strategies used by an organisation to protect data and information from threats.

In Area of Study 1 students apply the problem-solving stages of development and evaluation to develop their preferred design prepared in Unit 3, Area of Study 2, into infographics or dynamic data visualisations, and evaluate the solutions and project plan. Area of Study 1 forms the second part of the School-assessed Task (SAT). In Area of Study 2 students investigate security practices of an organisation. They examine the 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 | At least one data manipulation tool and one visualisation tool, for example database software, spreadsheet software, data visualisation software, tool for planning a project. |

A list of minimum software capabilities for database software, spreadsheet software and data visualisation software requirements will be published annually by the VCAA in the [*VCAA Bulletin*](https://www.vcaa.vic.edu.au/Pages/correspondence/index.aspx#bulletin).

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 or dynamic data visualisations that address a research topic or question by applying the problem-solving stages of development and evaluation. Details of the problem-solving methodology specifications are on [pages 12–15](#ProblemMethodology).

Effective designs and clarity of messages are key features of infographics and dynamic data visualisations, which are designed to communicate findings intended for a target audience. Students use software tools and functions of database and/or spreadsheet software and data visualisation software to support the types of data being manipulated to transform the designs into infographics or dynamic data visualisations.

Students evaluate the quality of their infographics 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 development of their project.

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

Outcome 1

On completion of this unit the student should be able to develop and evaluate infographics or dynamic data visualisations that present findings in response to a research question, and assess the effectiveness of the project plan in monitoring progress.

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

Key knowledge

*Digital systems*

* procedures and techniques for handling and managing files, including archiving, backing up, disposing of files and security
* the functional capabilities of software to create infographics and dynamic data visualisations

*Approaches to problem solving*

* characteristics of information for educating targeted audiences, including age appropriateness, commonality of language, culture inclusiveness and gender
* characteristics of efficient and effective infographics and dynamic data visualisations
* functions, techniques and procedures for efficiently and effectively manipulating data using software tools
* techniques for creating infographics and dynamic data visualisations
* techniques for validating and verifying data
* techniques for testing that solutions perform as intended
* techniques for recording the progress of projects, including adjustments to tasks and timeframes, annotations and logs
* strategies for evaluating the effectiveness of infographics and dynamic data visualisations solutions and assessing project plans.

Key skills

* monitor, modify and annotate project plans as necessary
* propose and implement procedures for managing files
* select and apply software functions, conventions, formats, methods and techniques to develop infographics or dynamic data visualisations
* select and apply data validation and testing techniques, making any necessary modifications
* apply evaluation criteria to evaluate the efficiency and effectiveness of infographics or dynamic data visualisations solutions
* assess the effectiveness of the project plan in managing the project.

Area of Study 2

Cybersecurity: data and information security

In this area of study students focus on data and information security and its importance to an organisation. Students investigate security strategies used by an organisation to manage the storage, communication and disposal of data and information in their networked environment. They examine the threats to this data and information, and evaluate the methods an organisation uses to protect their data and information. Students consider the consequences for an organisation that fails to protect their 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.

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 investigate the current data and information security strategies of an organisation, examine the threats to the security of data and information, and recommend strategies to improve current practices.

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

Key knowledge

*Digital systems*

* characteristics of wired, wireless and mobile networks
* types and causes of accidental, deliberate and events-based threats to the integrity and security of data and information used by organisations
* physical and software security controls for preventing unauthorised access to data and information and for minimising the loss of data accessed by authorised and unauthorised users
* the role of hardware, software and technical protocols in managing, controlling and securing data in information systems
* the advantages and disadvantages of using network attached storage and cloud computing for storing, communicating and disposing of data and information

*Data and information*

* characteristics of data that has integrity, including accuracy, authenticity, correctness, reasonableness, relevance and timeliness

*Interactions and impact*

* the importance of data and information to organisations
* the importance of data and information security strategies to organisations
* the impact of diminished data integrity in information systems
* key legislation that affects how organisations control the collection, storage, communication and disposal of their data and information: the *Health Records Act 2001*, the *Privacy Act 1988* and the *Privacy and Data Protection Act 2014*
* ethical issues arising from data and information security practices
* strategies for resolving legal and ethical issues between stakeholders arising from information security practices
* reasons to prepare for disaster and the scope of disaster recovery plans, including backing up, evacuation, restoration and test plans
* possible consequences for organisations that fail or violate security measures
* criteria for evaluating the effectiveness of data and information security strategies.

Key skills

* analyse and discuss the current data and information security strategies used by an organisation
* propose and apply criteria to evaluate the effectiveness of current data and information security strategies
* identify and evaluate threats to the security of data and information
* identify and discuss possible legal and ethical consequences of ineffective data and information security strategies
* recommend and justify strategies to improve current data and information security 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 timeframe.

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 *Advice for teachers* for this study, which includes 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** |
| **Unit 4****Outcome 2**Respond to a teacher-provided case study to investigate the current data and information security strategies of an organisation, examine the threats to the security of data and information, and recommend strategies to improve current practices. | **100** | The student’s performance will be assessed using one of the following:* structured questions
* a report in written format
* a report in multimedia format.
 |
| **Total marks** | **100****100** |  |

School-assessed Task

The student’s level of achievement in Unit 3, Outcome 2, and 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** |
| **Unit 3Outcome 2**Propose a research question, formulate a project plan, collect and analyse data, generate alternative design ideas and represent the preferred design for creating infographics or dynamic data visualisations. | A project plan (Gantt chart) indicating tasks, times, milestones, dependencies and critical path**AND**A collection of complex data sets that has been referenced**AND**An analysis that defines the requirements, constraints and scope of infographics or dynamic data visualisations**AND**A folio of alternative design ideas and detailed design specifications of the preferred design. |
| **Unit 4Outcome 1**Develop and evaluate infographics or dynamic data visualisations that present findings in response to a research question, and assess the effectiveness of the project plan in monitoring progress. | Infographics or dynamic data visualisations that present findings in response to a research question**AND*** an evaluation of the efficiency and effectiveness of infographics or dynamic data visualisations
* an assessment of the effectiveness of the project plan (Gantt chart) in monitoring project progress

in one of the following:* a written report
* an annotated visual plan.
 |

External assessment

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

Contribution to final assessment

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

End-of-year examination

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: two 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 and VCAL Administrative Handbook*](http://www.vcaa.vic.edu.au/pages/schooladmin/handbook/handbook.aspx).
* 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 a programming 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 and develop a set of working modules through the use of a programming language. Students examine a simple software requirements specification and a range of software design tools in order to apply specific processing features of a programming language to create working modules. In Area of Study 2 students analyse a need or opportunity, select an appropriate development model, 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 programming language. |
| Area of Study 2 | Unified Modelling Language (UML) and UML tools to create use cases. |

A list of suitable programming requirements will be published annually by the VCAA in the [*VCAA Bulletin*](https://www.vcaa.vic.edu.au/Pages/correspondence/index.aspx#bulletin). Schools select a language that fulfils these programming requirements.

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

|  |  |
| --- | --- |
| Area of Study 2 | Appropriate tool for documenting project plans. |

Area of Study 1

Software development: programming

In this area of study students examine the features and purposes of different design tools to accurately interpret the requirements and designs for developing working software modules. Students use a programming language and undertake the problem-solving activities of manipulation programming, validation, testing and documentation in the development stage.

The working modules do not have to be complete solutions and can focus on limited features of the programming language; however, students are expected to fully develop the working modules in accordance with the given designs. This will prepare students for creating a complete solution in Unit 4, Area of Study 1. Validation and testing techniques are applied to ensure modules operate as intended and internal documentation is written to explain the function of the modules. Students justify the use of the selected processing features and algorithms in the development of their working modules.

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

**Outcome 1**

On completion of this unit the student should be able to interpret teacher-provided solution requirements and designs, and apply a range of functions and techniques using a programming language to develop and test 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

*Data and information*

* characteristics of data types
* types of data structures, including associative arrays (or dictionaries or hash tables), one-dimensional arrays (single data type, integer index) and records (varying data types, field index)

*Approaches to problem-solving*

* methods for documenting a problem, need or opportunity
* methods for determining solution requirements, constraints and scope
* methods of representing designs, including data dictionaries, mock-ups, object descriptions and pseudocode
* formatting and structural characteristics of files, including delimited (CSV), plain text (TXT) and XML file formats
* a programming language as a method for developing working modules that meet specified needs
* naming conventions for solution elements
* processing features of a programming language, including classes, control structures, functions, instructions and methods
* algorithms for sorting, including selection sort and quick sort
* algorithms for binary and linear searching
* validation techniques, including existence checking, range checking and type checking
* techniques for checking that modules meet design specifications, including trace tables and construction of test data
* purposes and characteristics of internal documentation, including meaningful comments and syntax.

Key skills

* interpret solution requirements and designs to develop working modules
* use a range of data types and data structures
* use and justify appropriate processing features of a programming language to develop working modules
* develop and apply suitable validation, testing and debugging techniques using appropriate test data
* document the functioning of modules and the use of processing features through internal documentation.

Area of Study 2

Software development: analysis and design

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

Students prepare a project plan that includes student-determined and teacher-provided milestones that take into account all stages of the problem-solving methodology covered in this outcome and in Unit 4, Area of Study 1. Details of the relevant problem-solving methodology specifications are on [pages 12–15](#ProblemMethodology). Students justify the selection of an appropriate development model and monitor and modify their project plans. They do not have to use dedicated project-management software.

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

Students generate and document two or three design ideas for creating their solution. These could include annotations to indicate key functions and appearance. Evaluation criteria are developed and applied to select the preferred design idea. This design is then fully detailed, addressing the functionality and the user interface of the solution. These evaluation criteria will be 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 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 analyse and document a need or opportunity, justify the use of an appropriate development model, formulate a project plan, generate alternative design ideas and represent the preferred solution 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

*Digital systems*

* security considerations influencing the design of solutions, including authentication and data protection

*Data and information*

* techniques for collecting data to determine needs and requirements, including interviews, observation, reports and surveys

*Approaches to problem solving*

* functional and non-functional requirements
* constraints that influence solutions, including economic, legal, social, technical and usability
* factors that determine the scope of solutions
* features and purposes of software requirement specifications
* tools and techniques for depicting the interfaces between solutions, users and networks, including use case diagrams created using UML
* features of context diagrams and data flow diagrams
* techniques for generating design ideas
* criteria for evaluating the alternative design ideas and the efficiency and effectiveness of solutions
* methods of expressing solution designs using data dictionaries, mock-ups, object descriptions and pseudocode
* factors influencing the design of solutions, including affordance, interoperability, marketability, security and usability
* characteristics of user experiences, including efficient and effective user interfaces
* development model approaches, including agile, spiral and waterfall
* features of project management using Gantt charts, including the identification and sequencing of tasks, time allocation, dependencies, milestones and critical path

*Interactions and impact*

* goals and objectives of organisations and information systems
* key legal requirements relating to the ownership and privacy of data and information.

Key skills

* select a range of methods to collect and interpret data for analysis
* select and justify the use of an appropriate development model
* apply analysis tools and techniques to determine solution requirements, constraints and scope
* document an analysis as a software requirements specification
* generate alternative design ideas
* develop evaluation criteria to select and justify preferred designs
* produce detailed designs using appropriate design methods and techniques
* create, monitor and modify project plans using software.

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 timeframe.

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 *Advice for teachers* for this study, which includes 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** |
| **Unit 3****Outcome 1**Interpret teacher-provided solution requirements and designs, and apply a range of functions and techniques using a programming language to develop and test working software modules. | **100****100** | In response to teacher-provided solution requirements and designs, create working modules. |
| **Total marks** | **100****100** |  |

School-assessed Task

The student’s level of achievement in Unit 3, Outcome 2, and 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 46](#SAT2) of this study design.

External assessment

The level of achievement for Units 3 and 4 is also assessed by an end-of-year examination, which will contribute 50 per cent to the study score.

Unit 4: Software development

In this unit students focus on how the information needs of individuals and organisations are met through the creation of software solutions. They consider the risks to software and data during the software development process, as well as throughout the use of the software solution by an organisation.

In Area of Study 1 students apply the problem-solving stages of development and evaluation to develop their preferred design prepared in Unit 3, Area of Study 2, into a software solution and evaluate the solution, chosen development model and project plan. Area of Study 1 forms the second part of the School-assessed Task (SAT). In Area of Study 2 students examine the security practices of an organisation and the risks to software and data during the development and use of the software solutions. Students evaluate the current security practices and develop a risk management plan.

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 programming language. |

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

|  |  |
| --- | --- |
| Area of Study 1 | Appropriate tool for documenting project plans. |

A list of suitable programming requirements will be published annually by the VCAA in the [*VCAA Bulletin*](https://www.vcaa.vic.edu.au/Pages/correspondence/index.aspx#bulletin).

Area of Study 1

Software development: development and evaluation

In this area of study students develop the design they prepared in Unit 3, Area of Study 2, into a software solution that meets an identified need or opportunity by applying the problem-solving stages of development and evaluation. Details of the problem-solving methodology specifications are on [pages 12–15](#ProblemMethodology).

Appropriate processing features of a programming language, including validation, are used to develop an efficient and effective software solution. Testing techniques are used to ensure the software solution meets requirements.

Students prepare a usability test that addresses the core features of their software solution. The test must be undertaken by at least two potential ‘users’ and the results recorded. Potential ‘users’ could be students acting as real users of the solution. Students can make any necessary adjustments to their solution based on the results of their usability tests.

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 software solution using the evaluation criteria developed in Unit 3, Area of Study 2, and assess the effectiveness of their project plan and development model 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, evaluate the effectiveness of the development model 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

*Digital systems*

* procedures and techniques for handling and managing files and data, including archiving, backing up, disposing of files and data and security

*Data and information*

* ways in which storage media, transmission technologies and organisation of files affect access to data
* uses of data structures to organise and manipulate data

*Approaches to problem solving*

* processing features of a programming language, including classes, control structures, functions, instructions and methods
* characteristics of efficient and effective solutions
* techniques for checking that coded solutions meet design specifications, including construction of test data
* validation techniques, including existence checking, range checking and type checking
* techniques for testing the usability of solutions and forms of documenting test results
* techniques for recording the progress of projects, including adjustments to tasks and timeframes, annotations and logs
* factors that influence the effectiveness of development models
* strategies for evaluating the efficiency and effectiveness of software solutions and assessing project plans.

Key skills

* monitor, modify and annotate project plans as necessary
* propose and implement procedures for managing data and files
* develop a software solution and write internal documentation
* select and apply data validation and testing techniques, making any necessary modifications
* prepare and conduct usability tests using appropriate techniques, capture results, and make any modifications to solutions
* apply evaluation criteria to evaluate the efficiency and effectiveness of the software solution
* evaluate the effectiveness of the selected development model
* assess the effectiveness of the project plan in managing the project.

Area of Study 2

Cybersecurity: software security

Organisations are increasingly dependent on the use of software to achieve their goals and objectives. In this area of study students focus on the security risks to software and data during the software development process and throughout the use of the software solution by an organisation. Students analyse and evaluate the security of current software development practices, examine the risks to software and data, and consider the consequences of implementing software with ineffective security strategies. Physical and software controls, security vulnerabilities, web application and third-party software risks are investigated. Students recommend risk management strategies to improve current practices, taking into account the key legal requirements and ethical issues faced by an organisation.

Students apply systems thinking skills when analysing and evaluating software development security strategies within an organisation, and when recommending a risk management plan 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 examine the current software development security strategies of an organisation, identify the risks and the consequences of ineffective strategies and recommend a risk management plan to improve current security practices.

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

Key knowledge

*Digital systems*

* physical and software security controls used to protect software development practices and to protect software and data, including version control, user authentication, encryption and software updates
* software auditing and testing strategies to identify and minimise potential risks
* types of software security and data security vulnerabilities, including data breaches, man-in-the-middle attacks and social engineering, and the strategies to protect against these
* types of web application risks, including cross-site scripting and SQL injections
* managing risks posed by software acquired from third parties

*Data and information*

* characteristics of data that has integrity, including accuracy, authenticity, correctness, reasonableness, relevance and timeliness

*Interactions and impact*

* reasons why individuals and organisations develop software, including meeting the goals and objectives of the organisation
* key legislation that affects how organisations control the collection, storage (including cloud storage) and communication of data: the *Copyright Act 1968*, the *Health Records Act 2001*, the *Privacy Act 1988* and the *Privacy and Data Protection Act 2014*
* ethical issues arising during the software development process and the use of a software solution
* criteria for evaluating the effectiveness of software development security strategies
* the impact of ineffective security strategies on data integrity
* risk management strategies to minimise security vulnerabilities to software development practices.

**Key skills**

* analyse and discuss the current security controls to protect software development practices and to protect software and data
* identify and discuss the potential risks to software and data security with the current security strategies
* propose and apply criteria to evaluate the effectiveness of the current security practices
* identify and discuss the possible legal and ethical consequences to an organisation for ineffective security practices
* recommend and justify an effective risk management plan to improve current security 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 timeframe.

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 *Advice for teachers* for this study, which includes 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** |
| **Unit 4****Outcome 2**Respond to a teacher-provided case study to examine the current software development security strategies of an organisation, identify the risks and the consequences of ineffective strategies and recommend a risk management plan to improve current security practices. | **100****100** | The student’s performance will be assessed using one of the following:* structured questions
* a report in written format
* a report in multimedia format.
 |
| **Total marks** | **100****100** |  |

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** |
| **Unit 3Outcome 2**Analyse and document a need or opportunity, justify the use of an appropriate development model, formulate a project plan, generate alternative design ideas and represent the preferred solution design for creating a software solution. | A project plan (Gantt chart) indicating tasks, times, milestones, dependencies and critical path**AND**A justification of the selected development model as a written report**AND**An analysis that defines the requirements, constraints and scope of a solution in the form of a software requirements specification**AND**A folio of alternative design ideas and detailed design specifications of the preferred design. |
| **Unit 4Outcome 1**Develop and evaluate a software solution that meets requirements, evaluate the effectiveness of the development model and assess the effectiveness of the project plan. | A software solution that meets the software requirements specification**AND**Preparation and conduction of usability tests**AND*** an evaluation of the efficiency and effectiveness of the software solution
* an evaluation of the effectiveness of the selected development model
* an assessment of the effectiveness of the project plan (Gantt chart) in monitoring project progress

in one of the following:* a written report
* an annotated visual plan.
 |

External assessment

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

Contribution to final assessment

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

End-of-year examination

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: two 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 and VCAL Administrative Handbook*](https://www.vcaa.vic.edu.au/administration/vce-vcal-handbook/Pages/index.aspx.aspx).
* 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.