

**Copyright and Terms of Use Statement**

**© Australian Curriculum, Assessment and Reporting Authority 2022**

The material published in this work is subject to copyright pursuant to the Copyright Act 1968 (Cth) and is owned by the Australian Curriculum, Assessment and Reporting Authority (ACARA) (except to the extent that copyright is held by another party, as indicated).

The viewing, downloading, displaying, printing, reproducing (such as by making photocopies) and distributing of these materials is permitted only to the extent permitted by, and is subject to the conditions imposed by, the terms and conditions of using the ACARA website (see, especially, clauses 2, 3 and 4 of those terms and conditions). The terms and conditions can be viewed at [https://www.acara.edu.au/contact-us/copyright](https://aus01.safelinks.protection.outlook.com/?url=https%3A%2F%2Fwww.acara.edu.au%2Fcontact-us%2Fcopyright&data=04%7C01%7CSharon.Foster%40acara.edu.au%7C9931e11fa7684c603e6308d98331bbfb%7C6cf76a3aa824427092003d71673ec678%7C0%7C0%7C637685071906340874%7CUnknown%7CTWFpbGZsb3d8eyJWIjoiMC4wLjAwMDAiLCJQIjoiV2luMzIiLCJBTiI6Ik1haWwiLCJXVCI6Mn0%3D%7C1000&sdata=U5O4Vlbpf271IGmGiMh7fDwU4pLzzAiHpCQFylkp6s4%3D&reserved=0) 

TABLE OF CONTENTS

[CURRICULUM ELEMENTS 4](#_Toc95905117)

[Years 7–8 4](#_Toc95905118)

[Years 9–10 12](#_Toc95905119)

CURRICULUM ELEMENTS

Years 7–8

|  |
| --- |
| **Band level description** |
| By the end of Year 8 students should have had the opportunity to apply computational thinking by defining and decomposing real-world problems, creating user experiences, designing and modifying algorithms, and implementing them in a general-purpose programming language. This involves students practising problem decomposition, using approaches such as divide and conquer to more clearly understand a problem by describing its component parts. Students represent and communicate their algorithmic solutions using flowcharts and pseudocode. Students check their solutions meet the specifications by testing and debugging their algorithms before and during implementation. They develop a deeper understanding of abstraction by explaining how and why digital systems represent data as whole numbers, which are then represented in binary.  Students build on their skills from Mathematics (*Statistics*) in acquiring and interpreting data. In Digital Technologies, students continue to advance these skills and are also given opportunities to validate the data they acquire to ensure it is accurate and consistent. They collect and transform many types of data from a wide range of sources. Students model structured data in meaningful ways using spreadsheets and single-table databases, and analyse and visualise the data to extract meaning from it.  They apply design thinking by using divergent techniques, such as mind mapping, role-play and using graphic organisers, to generate design ideas for user experiences and solution designs. Students review these ideas against design criteria and created user stories throughout their implementation as general-purpose programming by assessing them against current and future needs. They extend the use of these design criteria and user stories to evaluate the future impact of existing solutions.  Students apply systems thinking by exploring the connections between hardware capabilities and tasks users want to perform. They investigate how data is transmitted via wired and wireless networks and explain the need for encryption to protect and secure data. Students use an increasing range of the features of digital tools to improve their efficiency and the consistency of the content they create, locate and communicate. They plan and manage projects individually and collaboratively, improving their control over the quality of their content. Students investigate personal security controls, including multi-factor authentication, to protect their data if passwords are compromised, and they understand the impact of phishing and other cyber security threats on people and data.  In Digital Technologies, students should have frequent opportunities for authentic learning by making key connections with other learning areas. |
| **Digital Technologies Achievement standard** |
| By the end of Year 8 students develop and modify creative digital solutions, decompose real-world problems, and evaluate alternative solutions against user stories and design criteria. Students acquire, interpret and model data with spreadsheets and represent data with integers and binary. They design and trace algorithms and implement them in a general-purpose programming language. Students select appropriate hardware for particular tasks, explain how data is transmitted and secured in networks, and identify cyber security threats. They select and use a range of digital tools efficiently and responsibly to create, locate and share content; and to plan, collaborate on and manage projects. Students manage their digital footprint. |

|  |
| --- |
| **Learning area Achievement standard** |
| By the end of Year 8 students explain how people design, innovate and produce products, services and environments for preferred futures. For each of the 4 prescribed technologies contexts students explain how the features of technologies impact on design decisions, and create designed solutions based on analysis of needs or opportunities. They acquire, interpret and model with spreadsheets and represent data with integers and binary. Students design and trace algorithms; and implement them in a general-purpose programming language. Students create and adapt design ideas, processes and solutions, and justify their decisions against developed design criteria that include sustainability. They communicate design ideas and solutions to audiences using technical terms and graphical representation techniques, including using digital tools. They select appropriate hardware for particular tasks, explain how data is transmitted and secured in networks, and identify cyber security threats. They use a range of digital tools to individually and collaboratively document and manage production processes to safely and responsibly produce designed or digital solutions for the intended purpose. Students manage their digital footprint. |

|  |  |  |
| --- | --- | --- |
| **Strand: Knowledge and understanding** | | **Years 7–8** |
| **Sub-strand: Digital systems** | | |
| **Content descriptions** *Students learn to:* | **Content elaborations**  *This may involve students:* | |
| explain how hardware specifications affect performance and select appropriate hardware for particular tasks and workloads  AC9TDI8K01 | * explaining how hardware specifications affect what, and how quickly, a digital system can perform tasks, for example how different bandwidth networks affect download speed and lag or how much random access memory (RAM) is needed for multimedia authoring * selecting appropriate hardware for particular tasks, for example choosing a powerful graphics card for computer gaming or large external storage for video editing * considering how First Nations Australian communities in areas classified as remote often share access to smartphone and internet services, and how the hardware specifications of these devices affect performance, for example where immediate and extended families share and access data through a single smartphone or device * explaining how the specifications of components in a system impact the speed with which AI models can be trained; for example, GPUs are more efficient at performing the mathematical calculations necessary for training generative AI than CPUs | |

|  |  |
| --- | --- |
| investigate how data is transmitted and secured in wired and wireless networks including the internet  AC9TDI8K02 | * describing physical networks and comparing their properties, for example the bandwidth, latency and reliability of wired versus wireless networks * explaining why cryptography is necessary for securing data, for example transmitting credit card details over the internet * exploring simple encryption and decryption algorithms, for example ROT13 and XOR * explaining how problems occur in network communication and how they can be solved, for example routers can drop packets and how Transmission Control Protocol (TCP) uses acknowledgements to confirm packets have been received |
| **Sub-strand: Data representation** | |
| investigate how digital systems represent text, image and audio data using integers  AC9TDI8K03 | * explaining how digital systems represent text as a sequence of individual characters numbered using the Unicode character set, for example upper-case and lower-case letters, punctuation and emoji * explaining how digital systems represent audio using whole numbers for the amplitude of the soundwave at a given sampling rate, for example -32,768 to 32,767 for 16-bit audio at 44,100 Hz * explaining how digital systems represent bitmap images (for example PNG and JPEG) as the colour of each pixel in separate red, green and blue (RGB) channels ranging from 0 to 255, and represent vector graphics, for example Scalable Vector Graphics (SVG) using the geometry of lines and shapes * investigating how a digital system converts audio data to integers as it records, stores and outputs sound, for example using the Welcome to Country app to understand the local history and Traditional Owners of the lands which students learn on to inform the programming of an Acknowledgement of Country in a local First Nations Australian language |
| explain how and why digital systems represent integers in binary  AC9TDI8K04 | * explaining how whole numbers can be represented in binary, for example counting in binary from 0 to 31, and recognising that one byte = 8 bits, which can represent from 0 to 255 * explaining how digital systems represent data in binary, for example by converting a character to its Unicode value, then converting that value into binary * explaining how circuits can perform binary operations represented as on/off states, for example showing how circuits with 2 switches can represent AND or OR gates |

|  |  |  |
| --- | --- | --- |
| **Strand: Processes and production skills** | | **Years 7–8** |
| **Sub-strand: Acquiring, managing and analysing data** | | |
| **Content descriptions** *Students learn to:* | **Content elaborations**  *This may involve students:* | |
| acquire, store and validate data from a range of sources using software, including spreadsheets and databases  AC9TDI8P01 | * acquiring data to answer questions for their own investigations, for example answering “Does the canteen sell the right food?” by designing a survey to collect food preferences data and accessing canteen sales data * judging how meaningful data is to a question, its correctness and how up to date the data is, for example “Does age affect the chance of cyclist injury?”, “Are self-reported accidents reliable?” and “Is the data before cycleways existed relevant?” * storing acquired data using specialised and general software appropriate for how it will be accessed and manipulated, for example a spreadsheet for visualisation or a pre-defined database for filtering and queries * acquiring, storing and validating data from a reputable source, such as the Australian Bureau of Statistics, to analyse the geographic distribution of First Nations Australians, with the aim to highlight past and emerging trends * ensuring that the data used to train an AI model minimises any potential biases in its output and is representative of the target audience; for example, training a model on data collected from a single demographic group may not produce correct outputs for a more diverse population | |
| analyse and visualise data using a range of software, including spreadsheets and databases, to draw conclusions and make predictions by identifying trends  AC9TDI8P02 | * summarising data based on its attributes to identify trends and make predictions, for example sorting crime data by type of offence, showing that burglaries have decreased over time to predict fewer burglaries will happen next year * visualising multidimensional data by choosing appropriate graphs, for example a scatter plot of food prices and sales, coloured by each food’s sugar content, or diagrams such as a social network diagram and maps of crime rates by location to reveal trends, outliers or other information * using an AI model with a natural language interface to generate queries to perform analysis; for example, describing a database schema and asking the model to generate an SQL query to find results that match a set of criteria * comparing the analysis performed by a trained predictive AI model with other analysis techniques; for example, comparing the output from a classification model against data tagged manually to verify its accuracy and effectiveness | |
| model and query the attributes of objects and events using structured data  AC9TDI8P03 | * modelling objects and events as structured data, that is, the attributes relevant to the task, for example products in the canteen and the sale of those products, with attributes such as the product name, price, quantity and nutritional value * using a spreadsheet table to model objects and events, including choosing appropriate formats for each column, and filtering and sorting rows to answer questions * interpreting and querying single-table databases using visual or simple SQL queries with SELECT, WHERE and ORDER BY clauses, for example answering queries in a database for a historical event | |

|  |  |
| --- | --- |
| **Sub-strand: Investigating and defining** | |
| define and decompose real-world problems with design criteria and by creating user stories  AC9TDI8P04 | * framing a problem in terms of what we know, why it is important and the outcome we want, for example matching the items in your fridge to possible recipes to reduce food waste * asking a series of questions and sub-questions to understand the problem and breaking it down into manageable parts, for example “How do we keep track of what items are in the pantry? Are there any dietary requirements that need to be considered?” * using a template such as “As a <type of user>, I want <some goal> so that <some reason>“, for example “As a user with a visual impairment I want to be able to get the news on my smartphone so that I can keep up with my world” * making predictions about future population distribution of First Nations Australians based on identified trends, for example analysing and visualising data using spreadsheets and databases on their population growth in metropolitan areas |
| **Sub-strand: Generating and designing** | |
| design algorithms involving nested control structures and represent them using flowcharts and pseudocode  AC9TDI8P05 | * designing an algorithm or modifying an existing algorithm to fix an error or change functionality, for example calculating the coins and notes needed for an amount of money and changing the algorithm to handle new denominations * describing algorithms precisely in pseudocode (structured English) or with flowcharts for each part of the problem, for example using separate flowcharts to describe the purchase of an item and the giving of change during the purchase * describing algorithms with nested control structures, including a nested if, for example IF it is raining THEN [IF parents are home THEN drive to school]; or an IF inside a loop, for example REPEAT [select the largest coin smaller than the remaining total, and subtract it] UNTIL the remainder is zero |
| trace algorithms to predict output for a given input and to identify errors  AC9TDI8P06 | * following an algorithm precisely to confirm it produces the expected output for the given input, for example desk check with a table of input, variables and output * specifying test cases and comparing the expected and actual output to determine the correctness of an algorithm, for example a test case of the change-calculating algorithm could have input $1.45 and expected output 1 x $1, 2 x 20c and 1 x 5c coins * following instructions for making woven baskets or nets by hand, as done by First Nations Australians, and making predictions of how the instructions would need to be modified to enable the item to be produced through automated manufacturing processes |
| design the user experience of a digital system  AC9TDI8P07 | * designing a user interface or experience to satisfy design criteria and user stories, using digital tools, for example sketch multiple pages of a website with wireframes, storyboards and simple branding guidelines for colours and styling * considering the factors of why a user might buy and use a product, in addition to its utility, for example how aligning the brand with the user’s values and identity contributes to its appeal * exploring the evolution of a user interface, for example comparing the design and branding of different search engines over time |
| generate, modify, communicate and evaluate alternative designs  AC9TDI8P08 | * reviewing and modifying a preferred design as part of the iterative development approach, for example making changes to overcome limitations of the design or better satisfy the user stories * using concept maps, wireframes or other diagrams to record and discuss the generated ideas, for example creating and discussing wireframes of a music streaming service, evaluating it against design criteria and user stories, such as the needs of diverse users * comparing multiple outputs from a generative model to determine the most suitable; for example, using AI tools to generate multiple prototypes of a user interface and selecting the design or features that best address users’ needs |
| **Sub-strand: Producing and implementing** | |
| implement, modify and debug programs involving control structures and functions in a general-purpose programming language  AC9TDI8P09 | * writing and editing programs to solve problems using branching, iteration, variables and functions in a general-purpose programming language, such as Python, JavaScript or C# * reading and interpreting an existing program and modifying the code to change functionality and fix errors, for example taking existing code for a weather forecasting app that includes temperatures and improving the output to include extra information such as rainfall, UV levels and air quality * writing a program that receives data from the environment to change the program behaviour, for example reading moisture level data from a soil sensor and switching on the watering system * writing a program that contains nested control structures to perform more complicated branching and decisions, for example using an IF statement inside a loop to count the warm days from an array containing temperature data only when the temperature for each day is more than 20 degrees Celsius * defining and using a function that produces different output based on the argument(s) it receives, for example a function that receives the name of an actor from user input, and searches a file or database to return a list of movies that actor appears in |

|  |  |
| --- | --- |
| **Sub-strand: Evaluating** | |
| evaluate existing and student solutions against the design criteria, user stories and possible future impact  AC9TDI8P10 | * evaluating how an existing solution ensures users can control their safety and experience online as described in the Safety by Design Vision for Young People, for example ensuring privacy settings are comprehensive, easy to use and set to maximum protection by default * reviewing the requirements of a user story to ensure that their solution meets the user’s needs, for example making sure that recommendations offered by their music application are of a similar genre to the rest of the user’s library * judging existing solutions on the basis of their possible impact on the economy, environment or society, for example cloud computing services decrease data loss but require vast amounts of electricity to power the servers * discussing the risks and consequences of AI-generated content on social media platforms; for example, the potential for the spread of misinformation due to high volumes of automatically generated and intentionally misleading content being posted |
| **Sub-strand: Collaborating and managing** | |
| select and use a range of digital tools efficiently, including unfamiliar features, to create, locate and communicate content, consistently applying common conventions  AC9TDI8P11 | * locating relevant content from multiple sources, exploring advanced search functions and targeted criteria, for example using specific filters such as date range, image size, file type and usage licence * selecting and using appropriate digital tools, for example when participating in online lessons or planning sessions using a common video conferencing tool * applying common conventions consistently when creating content, for example organising content in paragraphs and within a heading hierarchy, writing captions to describe images and using gender-inclusive pronouns, where appropriate * creating logical storage locations for project assets and resources together with an outline to ensure collaborators are up to date, for example creating a logical storage area for a group to share content and ideas in a timely way about the canteen issue they are solving * using effective prompts with generative AI models to create output that is better suited to the problem being solved; for example, specifying the voice, tone and brevity for a persuasive news article with a restrictive word limit * using a progressive series of prompts with generative models to refine output to improve its correctness; for example, performing translation from one language to another and instructing the model to correct errors in translation |
| select and use a range of digital tools efficiently and responsibly to share content online, and plan and manage individual and collaborative agile projects  AC9TDI8P12 | * collaborating effectively online using cloud storage, for example setting up and managing a shared space in an online repository to co-develop content for an app which presents and checks safety aspects of working in a specific setting such as a kitchen, lab, workshop or greenhouse * displaying empathy for diverse cultural expectations when participating in teams and in online communities, for example showing sensitivity around images or names of deceased people, and valuing the intellectual property and perspectives of others * demonstrating agile project management skills and understanding, for example when collaborating with First Nations Australians’ community groups to develop digital solutions to projects: following cultural protocols, including relevant permissions and attributions; acknowledging diversity, capability and strength; and addressing risks and responsibilities such as privacy, security and accuracy of data * determining and recording the tasks, responsibilities and timeframes for a collaborative project, for example using a spreadsheet to record tasks and their sequence, critical dates and who is responsible for each task so a project can be finished on time * using AI tools to decompose high-level instructions into more detailed steps to assist with completing a task; for example, asking an AI model to break down the steps involved in building a website from scratch |
| **Sub-strand: Privacy and security** | |
| explain how multi-factor authentication protects an account when the password is compromised and identify phishing and other cyber security threats  AC9TDI8P13 | * explaining how multi-factor authentication prevents unauthorised access by prompting the account owner for a token or single-use password, for example demonstrating how a funds transfer from their bank account requires not only logging in, but provision of a one-time password received via SMS * identifying the common techniques used in phishing scams to identify and exploit susceptible users, for example using an email address from an unofficial domain when pretending to be an online retailer, or including grammatical errors to help filter out users who are more likely to detect the scam |
| investigate and manage the digital footprint existing systems and student solutions collect, and assess if the data is essential to their purpose  AC9TDI8P14 | * investigating the ethical obligations of individuals and organisations regarding ownership and privacy of data and information by researching an online platform’s privacy policy for data collection, use and storage information and discussing impacts on digital footprint * reviewing and managing their digital footprint across online digital tools that they use, for example selecting their default privacy and sharing settings on social media accounts * investigating how recommendation algorithms used in media services rely on data that tracks user habits, for example how music streaming services generate playlists that contain songs from genres and artists that are similar to those you listen to regularly * assessing the appropriateness and relevance of data collected by surveys from other students and organisations they complete online, for example identifying that providing your address data is not necessary for a survey asking about your food preferences but providing the address for the Census would be appropriate * explaining the risks associated with sharing personal data due to the ease with which generative AI models can create new content; for example, from short videos and audio recordings it is possible for convincing deep fake videos to be generated and distributed for malicious purposes |

Years 9–10

|  |
| --- |
| **Band level description** |
| By the end of Year 10 students should have had the opportunity to apply computational thinking by defining and decomposing real-world problems, creating user experiences, designing and modifying algorithms, and implementing them, including in an object-oriented programming language. Students use techniques, including interviewing stakeholders to develop user stories, to increase the precision of their problem definitions and solution specifications. They verify their solutions solve the problem by validating their algorithms, represented as flowcharts and pseudocode, and using test cases to confirm the correctness of their solutions. Students develop their object-oriented programming skills, and apply them to develop, modify and debug programs. They explain the importance of abstraction by representing online documents in terms of content, structure and presentation, as well as exploring simple data compression techniques and comparing their effectiveness.  Students consolidate their skills in data acquisition and interpretation, cleaning and validating data to ensure it is accurate, consistent and domain appropriate. They model multidimensional data in more complex spreadsheets and relational databases, filtering and querying it to give insights into its meaning, and to pose further questions or make conclusions. They visualise this data in customisable ways, allowing greater exploration of trends and outliers to support or challenge their analyses.  Students apply design thinking by using divergent techniques to generate design ideas for user experiences and solutions. They filter and prototype these ideas, developing user stories and applying design criteria based on current and future needs and enterprising opportunities, as well as their created user stories, and revise and further develop their preferred ideas based on their analysis. Students extend on these design criteria and user stories to evaluate the enterprise opportunities and future impact of existing solutions.  Students consolidate their systems thinking by exploring how the hardware and software components of digital systems interact to manage, control and secure access to data. They increasingly use advanced features of existing and emerging digital tools to create interactive content for a diverse audience. They explore simple tools that help plan tasks, timelines and responsibilities for individual and collaborative projects. Students extend their knowledge of the importance of security by developing cyber security threat models and exploring an example of a supply chain vulnerability. They critique the digital footprint created by existing systems and their own solutions by applying the Australian Privacy Principles.  In Digital Technologies, students should have frequent opportunities for authentic learning by making key connections to other learning areas. |
| **Achievement standard** |
| By the end of Year 10 students develop and modify innovative digital solutions, decompose real-world problems, and critically evaluate alternative solutions against stakeholder elicited user stories. Students acquire, interpret and model complex data with databases and represent documents as content, structure and presentation. They design and validate algorithms and implement them, including in an object-oriented programming language. Students explain how digital systems manage, control and secure access to data; and model cyber security threats and explore a vulnerability. They use advanced features of digital tools to create interactive content, and to plan, collaborate on and manage agile projects. Students apply privacy principles to manage digital footprints. |

|  |  |  |
| --- | --- | --- |
| **Strand: Knowledge and understanding** | | **Years 9–10** |
| **Sub-strand: Digital systems** | | |
| **Content descriptions** *Students learn to:* | **Content elaborations**  *This may involve students:* | |
| investigate how hardware and software manage, control and secure access to data in networked digital systems  AC9TDI10K01 | * explaining how the operating system hides the complexity of different hardware from applications, for example applications can treat input from a mouse and touch screen in the same way * exploring how public key cryptography, for example TLS, and hashing, such as SHA-1, secure the storage and transmission of data * configuring a simple network using real or simulated hardware and observing packets moving around the network, for example monitoring packets on simulated switches and networked devices * explaining how domain names and IP addresses allow data to be transmitted to specific networked devices, for example DNS and routing tables * describing elements of access control and explaining why they are necessary, for example authentication and permissions for restricting access to install software to administrators | |
| **Sub-strand: Data representation** | | |
| represent documents online as content (text), structure (markup) and presentation (styling) and explain why such representations are important  AC9TDI10K02 | * representing documents by separating the content (the text in the document), the structure (the document structure such as headings and paragraphs) and presentation (how the document is laid out and styled) * writing webpages using HyperText Markup Language (HTML) for the content and structure and Cascading Style Sheets (CSS) for styling the page and explaining how HTML tags separate content from structure * explaining how representing content, structure and presentation separately allows each of them to be designed, edited, manipulated and stored independently of the others and why this is important | |
| investigate simple data compression techniques  AC9TDI10K03 | * using an algorithm to identify patterns in data and represent them in a compressed way, for example repeated pixels in an image with run-length encoding * exploring the difference between lossy and lossless compression and the consequences of each, for example exploring codecs for audiovisual compression such as MP3, MP4 and WAV formats, considering energy requirements of file sizes * examining an image and discussing whether the image quality would be compromised if all the blue pixels of the sky in one row were to be replaced by one token and the number of pixels it represents | |
| **Strand: Processes and production skills** | | **Years 9–10** |
| **Sub-strand: Acquiring, managing and analysing data** | | |
| **Content descriptions** *Students learn to:* | **Content elaborations**  *This may involve students:* | |
| develop techniques to acquire, store and validate data from a range of sources using software, including spreadsheets and databases  AC9TDI10P01 | * developing systems that store structured data, for example a movie or travel review website that collects Likert scale ratings and written reviews * developing systems that check data is correct and meaningful using automated techniques and manual analysis, for example, validating movie review data using rules and user interface elements, and detecting bias and fake reviews through simple statistical analysis * developing systems that acquire, use and protect data according to the Australian Privacy Principles, for example ensuring personally identifiable information is not publicly shared without consent and is protected from unauthorised access * accessing and storing data from the Australian Bureau of Statistics in a format that is useful for analysis, for example acquiring data on population growth across age groups in Australia * identifying strengths and weaknesses of collecting data using different methods, for example online surveys, face-to-face interviews, phone interviews, observation, comments in response to a social media posting, phone logs, browser history and online webcam systems * considering how training data issues such as the zero problem dictate the output from predictive models; for example, a model with many examples of horses and no zebras in its training data is likely to classify all zebras as horses | |
| analyse and visualise data interactively using a range of software, including spreadsheets and databases, to draw conclusions and make predictions by identifying trends and outliers  AC9TDI10P02 | * summarising data, its attributes and the relationships between data sets, identifying trends and outliers to draw conclusions and make predictions, for example summarising data about electorates and their demographics, historical swings and exceptions to predict an election outcome * developing interactive visualisations for exploring complex data, for example population, life expectancy and fertility rate in motion charts * using software to visualise and compare data to identify patterns, relationships and trends, for example investigating emerging trends in Australia’s industries * exploring machine learning, a form of artificial intelligence where an algorithm is trained using a data set, for example to classify images into categories * adjusting parameters of an AI model to observe the impact of different factors on predicted outcomes; for example, changing the weighting of different input variables to see how much it changes the model's outputs | |
| model and query entities and their relationships using structured data  AC9TDI10P03 | * modelling entities and processes, their attributes and the relationships between them, for example creating database tables for a movie, a user and their movie review, where a movie has a title, genre and release date, and a review has a movie, a user and their rating and comments * using structured data to help in decision-making, for example creating a data schema for a relational database and building the database, incorporating query and reporting functionality to solve a problem of student choice * interpreting and querying multi-table databases using SQL queries with SELECT, WHERE and simple JOIN/GROUP BY clauses and counting, for example checking that each user has only reviewed each movie once | |
| **Sub-strand: Investigating and defining** | | |
| define and decompose real-world problems with design criteria and by interviewing stakeholders to create user stories  AC9TDI10P04 | * creating user stories by interviewing a stakeholder to complete a template such as “As a <type of user>, I want <some goal> so that <some reason>”, for example interviewing an amateur athlete to complete the template, such as “As an athlete, I want to ensure my energy intake is high enough to complete a half marathon.” * defining the problem with precision and some awareness of scope, for example “How do we encourage people to balance their energy intake and expenditure?” and “Can this be solved in a specified timeframe?” * asking questions that help them define the problem more precisely, for example “How do we measure energy intake?” or “How much energy does each exercise expend?” * recognising the importance of diverse perspectives when defining the problem and devising survey or interview questions to elicit stakeholder needs, for example “What types of exercise count?” and allowing open-ended responses to the exercise they do * exploring how First Nations Australian cultural stories and languages are being preserved with digital systems, for example how communities could record, animate and maintain their connections with culture and language in a contemporary format that resonates with young people to help ensure that vital practices continue | |
| **Sub-strand: Generating and designing** | | |
| design algorithms involving logical operators and represent them as flowcharts and pseudocode  AC9TDI10P05 | * designing an algorithm or modifying an existing algorithm to fix, extend or improve it, for example fixing a bug in an algorithm to detect if 2 shapes intersect when the shapes just touch or extending the algorithm to support a new shape * describing algorithms using flowcharts or other appropriate diagram types, for example a decision tree for classifying an animal based on physical characteristics * describing algorithms precisely and succinctly using pseudocode, for example short, unambiguous statements such as IF length of word is greater than 4 AND first letter is a vowel * using Boolean operations (that is, AND, OR and NOT) to express complex conditions in control structures, for example IF [the temperature is above 30 degrees AND people are inside the building] THEN open the windows | |
| validate algorithms and programs by comparing their output against a range of test cases  AC9TDI10P06 | * tracing and debugging an algorithm by identifying when its state is unexpected, why this has occurred, and the changes needed to correct it, for example identifying that a loop has finished one iteration too early * determining boundary test cases and testing that they are handled correctly by a program, for example checking that an intersection is detected when 2 shapes are perfectly aligned * generating invalid input and user errors and testing that they are handled appropriately by a program, for example checking that a program does not crash when text is entered instead of a number | |
| design and prototype the user experience of a digital system  AC9TDI10P07 | * designing engaging user experiences, considering aesthetics, functionality and the feeling of enjoyment and satisfaction of the user * prototyping a user experience, using simple graphical tools that support clicking on an image to change slides or views, for example using a presentation tool or a no-code user interface prototyping tool to design a simple mobile app * considering all aspects of a product as perceived by the users, for example evaluating users’ initial experience of setting up and using a system, or users’ emotional or cultural response to using a digital system * designing documentation, branding and marketing for a digital solution, for example a product demonstration screencast or ‘getting started’ user guide | |
| generate, modify, communicate and critically evaluate alternative designs  AC9TDI10P08 | * eliminating design ideas by evaluating them against the design criteria and user stories, for example in consultation with stakeholders, reviewing the design ideas, making modifications if necessary, and further developing the design of the preferred solution * using a range of ideation techniques to create multiple design ideas for a solution, for example using graphic organisers, role-play and mind mapping to develop and then record a range of ideas without evaluating them first * combining the output from generative AI models and human capital from recognised experts to meet a specific need; for example, using a range of outputs from an image generator as inspiration for modelling a 3D character in a game | |

|  |  |
| --- | --- |
| **Sub-strand: Producing and implementing** | |
| implement, modify and debug modular programs, applying selected algorithms and data structures, including in an object-oriented programming language  AC9TDI10P09 | * writing and editing programs to solve problems using algorithms and data structures in general-purpose and object-oriented programming languages, such as Python, JavaScript or C# * debugging a program by locating an error, modifying the program, and verifying that the changes made are correct, for example identifying the line in the code where an error occurs by reading an error message, printing out the variables to deduce what is causing the problem, and testing any fixes by entering data that could create similar errors * reading and interpreting programs split across files, functions or classes, and modifying them to add functionality, for example using the codebase of an existing adventure game and building new characters, levels or abilities * writing programs that receive structured data from the user that determines the program’s behaviour, for example processing a file that contains timestamped data captured by an altitude sensor from a plastic bottle rocket to graph its flight path * applying simple data structures and algorithms appropriately in their programs, for example using an array to store temperature data for a month, a dictionary to store character information in a role-playing game (RPG), and the binary search function from a library to find a value in a sorted array * defining their own classes to model and define the actions that can be performed on data in their programs, for example defining a class for a book that stores information such as the author, title and publisher, and methods that are used to track the book’s status in a library management system or store inventory * selecting different types of data structures such as array, record and object to model structured data |
| **Sub-strand: Evaluating** | |
| evaluate existing and student solutions against the design criteria, user stories, possible future impact and opportunities for enterprise  AC9TDI10P10 | * evaluating how an existing solution detects violations of site rules and imposes sanctions as described in the Safety by Design Vision for Young People, for example enforcing rules using artificial intelligence and human moderators to detect inappropriate behaviour, and consistently imposing consequences * judging the quality of the output of their solution against the design criteria, for example confirming that the stock levels recorded by their inventory management application are accurate within allowed parameters * evaluating how existing products have pivoted to meet the needs of a different set of users, for example how new social media applications continue to expand the types of media people share and the interactions they have online * examining the unintended consequences of an image generation solution implemented using AI; for example, using a model trained on a homogenous population to generate graphics that do not represent the diversity of customers in a website’s target audience * considering the complexities associated with training predictive models to capture events that occur with low probability; for example, in training data for an autonomous vehicle, including a person lying on the road and the correct behaviour for that situation |
| **Sub-strand: Collaborating and managing** | |
| select and use emerging digital tools and advanced features to create and communicate interactive content for a diverse audience  AC9TDI10P11 | * locating relevant content by using Boolean operators or artificial intelligence search features of search engines and desktop search, for example queries with ‘and’ and ‘not’ or advanced image similarity search * using emerging technologies to add dynamic features to content, for example using a QR code to trigger an augmented reality (AR) overlay of how to use a tool safely in the workshop * incorporating interactive elements into content to assist with analysis, for example adding sliders to visualisations to allow the user to control and view time series data * ensuring content is accessible by using built-in accessibility features, for example using ALT tags in images inside HTML to ensure screen readers can communicate content for people who are visually impaired * combining the output from multiple generative AI sources to communicate a complex idea or narrative; for example, using images, sounds and text from a variety of tools to produce an interactive animation |
| use simple project management tools to plan and manage individual and collaborative agile projects, accounting for risks and responsibilities  AC9TDI10P12 | * managing collaborative projects using appropriate platforms, for example sharing code development using a collaborative version control tool such as GitHub to plan, decompose and manage a project with their peers * establishing clear lines of responsibility and tasks for all members of a project team, for example assigning roles to all team members and using a spreadsheet to sequence tasks and track progress to ensure all work is completed by a specified deadline * using relevant legislation or guidelines to inform their solutions, for example correctly controlling and treating data collected from users by applying techniques that ensure data protection, privacy and copyright requirements are followed * planning the creation of content using project management tools to mitigate potential risks or project delays, for example incorporating project management techniques including scheduling, revision, iteration and evaluation into common collaboration tools to ensure timely delivery of effective solutions * accounting for appropriate project management responsibilities, for example when collaborating with First Nations Australians’ community groups to develop digital solutions to projects: following cultural protocols, including relevant permissions and attributions; acknowledging diversity, capability and strength; and addressing risks and responsibilities such as privacy, security and accuracy of data * incorporating suggestions made by built-in virtual assistants in project planning and organisation tools to streamline and prioritise work; for example, using AI to summarise meeting outcomes, identify important tasks and forecast project risks |
| **Sub-strand: Privacy and security** | |
| develop cyber security threat models, and explore a software, user or software supply chain vulnerability  AC9TDI10P13 | * using a data flow diagram to understand how private information moves through a system and when it would be the most likely target of a cyber attack, for example mapping how data moves between the user and server when using a web application, and identifying that sending the data in plaintext would make it susceptible to a man-in-the-middle attack * exploring the impact of a cyber security threat by systematically following the steps involved in bypassing a known vulnerability in their own software, for example manually changing the value stored in a login cookie to that of another user to observe the impact of unauthorised access on the system |
| apply the Australian Privacy Principles to critique and manage the digital footprint that existing systems and student solutions collect  AC9TDI10P14 | * critiquing the extent to which online services allow them to control access to their data in line with the Australian Privacy Principles, for example assessing whether their social media accounts allow them to update their contact information if these details change, and who else can see that information on the platform * using the Australian Privacy Principles as a reference to evaluate the steps they are taking to protect user information in their application, for example explaining how they are storing passwords using cryptographic hashing algorithms so that a data breach does not expose their users to security vulnerabilities due to password re-use |