



Technologies: Digital Technologies

Teaching, learning and assessment exemplar

Year 9

Privacy chatbot



Acknowledgement of Country

Kaya. The School Curriculum and Standards Authority (the Authority) acknowledges that our offices are on Whadjuk Noongar boodjar and that we deliver our services on the country of many traditional custodians and language groups throughout Western Australia. The Authority acknowledges the traditional custodians throughout Western Australia and their continuing connection to land, waters and community. We offer our respect to Elders past and present.

Background

This teaching, learning and assessment exemplar (the exemplar) has been developed by the School Curriculum and Standards Authority (the Authority) as part of the *School Education Act Employees (Teachers and Administrators) General Agreement 2017* (Clause 61.1–61.3).

Copyright

© School Curriculum and Standards Authority, 2026

This document – apart from any third-party copyright material contained in it – may be freely copied, or communicated on an intranet, for non-commercial purposes in educational institutions, provided that the School Curriculum and Standards Authority (the Authority) is acknowledged as the copyright owner, and that the Authority's moral rights are not infringed.

Copying or communication for any other purpose can be done only within the terms of the *Copyright Act 1968* or with prior written permission of the Authority. Copying or communication of any third-party copyright material can be done only within the terms of the *Copyright Act 1968* or with permission of the copyright owners.

Any content in this document that has been derived from the Australian Curriculum may be used under the terms of the [Creative Commons Attribution 4.0 International licence](#).

Disclaimer

Any resources such as texts, websites and so on that may be referred to in this document are provided as examples of resources that teachers can use to support their learning programs. Their inclusion does not imply that they are mandatory or that they are the only resources relevant to the course. Teachers must exercise their professional judgement as to the appropriateness of any they may wish to use.

This resource utilises electronic web-based resources, such as videos and image galleries. Teachers should be present while an electronic resource is in use and close links immediately after a resource, such as a video has played to prevent default 'auto play' of additional videos. Where resources are referred for home study, they should be uploaded through Connect, or an equivalent system, that filters advertising content.

Contents

The Western Australian Curriculum	1
The Technologies curriculum	1
This exemplar.....	2
Catering for diversity.....	2
Using this exemplar.....	3
Links to electronic resources.....	3
Best practice	4
Teaching and learning	4
Assessing	4
Reflecting.....	4
Privacy chatbot	5
Year level description	6
Achievement standard	7
Lessons 1–15	9
Appendix A	39
Appendix B	45
Acknowledgements.....	60

The Western Australian Curriculum

The *Western Australian Curriculum and Assessment Outline* (the *Outline* – <https://k10outline.scsa.wa.edu.au/>) sets out the mandated curriculum, guiding principles for teaching, learning and assessment, and support for teachers in their assessment and reporting of student achievement. The *Outline* recognises that all students in Australian schools, or international schools implementing the Western Australian Curriculum, are entitled to be given access to the eight learning areas described in the *Alice Springs (Mparntwe) Education Declaration*, December 2019.

The Technologies curriculum

The mandated curriculum is presented in the year level syllabus documents.

The Technologies curriculum delivers a sequential and age-appropriate progression of learning with the following key elements:

- a year level description that provides an overview of the context for teaching and learning in the year
- a series of content descriptions, populated through strands and sub-strands, that sets out the knowledge, understanding and skills that teachers are expected to teach and students are expected to learn
- an achievement standard that describes an expected level that the majority of students are achieving by the end of a given year of schooling. An achievement standard describes the quality of learning (e.g. the depth of conceptual understanding and the sophistication of skills) that would indicate the student is well placed to commence the learning required in the next year.

This exemplar

This Technologies exemplar articulates the content in the *Outline* and approaches to teaching, learning and assessment reflective of the Principles of Teaching, Learning and Assessment.

This exemplar demonstrates a sequence of teaching and learning, including suggested assessment points, for 15 lessons.

Catering for diversity

This exemplar provides a suggested approach for the delivery of the curriculum and reflects the rationale, aims and content structure of the learning area. When planning the learning experiences, consideration has been given to ensuring that they are inclusive and can be used in, or adapted for, individual circumstances. It is the classroom teacher who is best placed to consider and respond to (accommodate) the diversity of their students. Reflecting on the learning experiences offered in this exemplar will enable teachers to make appropriate adjustments (where applicable) to better cater for students' gender, personal interests, achievement levels, socio-economic, cultural and language backgrounds, experiences and local area contexts.

Using this exemplar

This teaching, learning and assessment exemplar provides suggestions to support the delivery of the mandated curriculum content. The exemplar provides:

- a teaching and learning sequence
- the mandated curriculum content to be taught at each point of the teaching and learning sequence, suggested resources, a sample assessment task and marking key
- the number of lessons to deliver the teaching and learning experiences
- learning intentions and support notes that may provide focus questions and additional information and/or examples to assist with the interpretation of curriculum content
- support notes to assist teachers to unpack the content and support teaching and learning experiences
- teaching and learning experiences that outline the structure of the lesson. These explicitly state each activity that the lesson will progress through and the key focus area for that activity.

Links to electronic resources

This sequence of lessons may utilise electronic web-based resources, such as videos and image galleries. Teachers should be present while an electronic resource is in use and close links immediately after a resource, such as a video, has played to prevent default 'auto play' of additional videos. Where resources are referred for home study, they should be uploaded through Connect, or an equivalent system, that filters advertising content.

Best practice

Teaching and learning

The teaching and learning opportunities offered in this exemplar are not exhaustive. Thus, teachers are encouraged to make professional decisions about which learning experiences, and the sequence in which they are delivered, are best suited to their classroom context, taking into account the availability of resources and student ability.

This sample may prove a useful starting point for amplifying creativity in the classroom, while presenting the embedded expectations of the Western Australian Curriculum: Technologies.

Teachers may find opportunities to incorporate the General Capabilities and the Cross-curriculum Priorities into the teaching and learning program.

Ways of teaching – teachers can locate additional information on the Ways of teaching from the School Curriculum and Standards Authority (the Authority) website

<https://k10outline.scsa.wa.edu.au/home/wa-curriculum/learning-areas/technologies/digital-technologies/p-10-digital-technologies-teaching/digital-technologies-ways-of-teaching>.

Assessing

Assessment, both formative and summative, is an integral part of teaching and learning. Assessment should arise naturally out of the learning experiences provided to students. In addition, assessment should provide regular opportunities for teachers to reflect on student achievement and progress. As part of the support it provides for teachers, this exemplar includes suggested assessment points. It is the teacher's role to consider the contexts of their classroom and students, the range of assessments required, and the sampling of content descriptions selected to allow their students the opportunity to demonstrate achievement in relation to the year level achievement standard. Teachers are best placed to make decisions about whether the suggested assessment/s are used as formative or summative assessment and/or for moderation purposes.

Ways of assessing – a range of assessment strategies that can enable teachers to understand where students are in their learning is available on the Authority website

<https://k10outline.scsa.wa.edu.au/home/wa-curriculum/learning-areas/technologies/digital-technologies/p-10-digital-technologies-assessing/digital-technologies-ways-of-assessing>.

Reflecting

Reflective practice involves a cyclic process during which teachers continually review the effects of their teaching and make appropriate adjustments to their planning. The cycle involves planning, teaching, observing, reflecting and replanning.

This exemplar supports reflective practice and provides flexibility for teachers in their planning. The exemplar shows how content can be combined and revisited throughout the year. Teachers will choose to expand or contract the amount of time spent on developing the required understandings and skills according to their reflective processes and professional judgements about their students' evolving learning needs.

Privacy chatbot

This exemplar can be used to develop students' understanding of digital systems, the *Australian Privacy Act 1988* and the creation of digital solution.

If the suggested learning experiences and the relevant Western Australian Curriculum content for this unit have been studied, students will be well positioned to address the requirements of the assessment task to the best of their ability.

Students create a digital solution to demonstrate design of algorithms, represented diagrammatically and in structured English, and validate plans and programs through tracing. They will implement and apply data storage and organisation techniques. Students create and use interactive solutions for sharing ideas and information online, taking into account social contexts.

Working individually and collaboratively, students produce a Privacy chatbot to test the user's knowledge of the *Australian Privacy Act 1988*, including the 13 Australian Privacy Principles, and how these apply to electronic transmission of personal data.

Year level description

In the middle adolescence phase of schooling, teaching and learning programs encourage students to develop an open and questioning view of themselves as active participants in their society and the world.

Digital Technologies further develops student understanding and skills in computational thinking, such as the precise and accurate describing of problems, and the use of functions to create solutions. It also focuses on engaging students with specialised learning to further develop their critical and design thinking skills.

In Year 9, students consider the role of hardware and software in securing the movement of data in a digital system. Students explore different methods of manipulation, storage and transmission of data, and students develop abstractions, identifying standard elements and modelling entities and their relationships. They interrogate security practices and techniques used to compress data and learn about the importance of separating content, presentation and behavioural elements for data integrity and maintenance purposes.

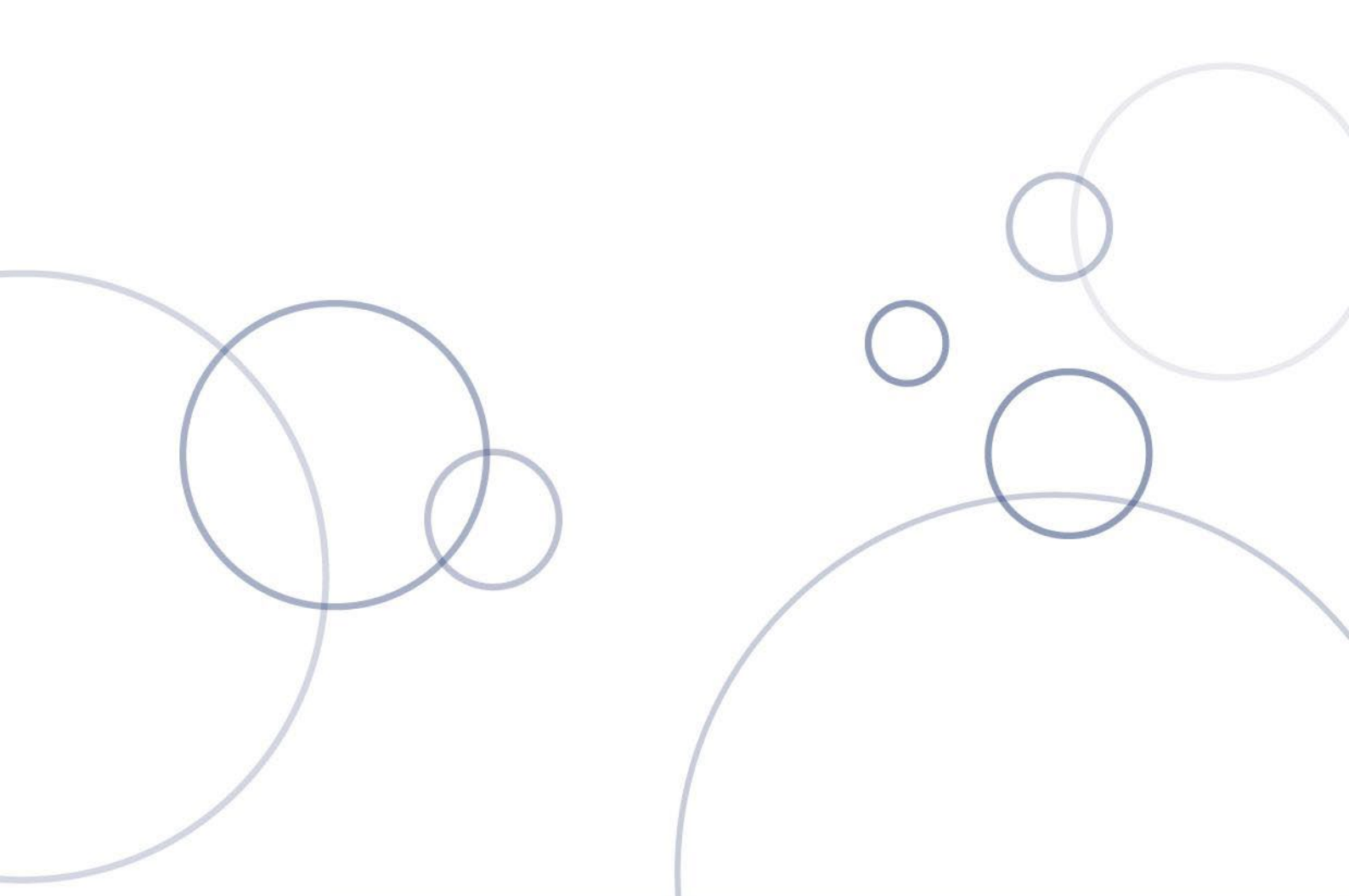
When defining problems, students consider the functional and non-functional requirements of a solution through interacting with users and reviewing processes. They consolidate their algorithmic design skills to incorporate testing. Students develop solutions to problems and evaluate their solutions and existing information systems, based on a set of criteria. They consider the privacy and security implications of how data is used and controlled, and suggest how policies and practices can be improved to ensure the sustainability and safety of information systems.

Achievement standard

By the end of the year:

Students describe the role of hardware and software to secure the movement of data in digital systems with a focus on cybersecurity threat models. They identify different methods used for manipulation, storage and transmission of data. Students acquire, store and validate data from a range of sources using software, including spreadsheets and databases. They define and decompose real-world problems by interviewing stakeholders to develop user experiences. Students design and prototype the user experience of a digital system. They design algorithms that use functions and represent them as flow charts and/or pseudocode. Students implement and modify programs in a general-purpose programming language and trace algorithms to predict output against a range of test cases. They describe the Australian Privacy Principles regarding the collection and ownership of data.

In Digital Technologies, students identify and define the needs of a stakeholder to develop a design brief for a solution. They investigate a range of components and resources to develop ideas and identify and consider constraints. Students apply design thinking, creativity and enterprise skills to produce purposeful and holistic solutions. Students select, test and implement appropriate technologies and processes to produce effective solutions and evaluate design processes against student-developed criteria. Students manage projects using digital technologies through an agile and collaborative approach, and consider time, risk, economic and sustainable factors.



Lessons 1–15

Privacy chatbot

Lesson 1: Artificial intelligence

The Western Australian Curriculum content addressed in this lesson is below.

Digital systems

- Role of hardware and software to manage, control and secure the movement of data in a digital system

Design thinking skills

Investigating and defining

- Investigate a range of technologies, resources and/or components to develop ideas and solutions, with consideration of social, ethical and other constraints
-

Lesson outline

Learning intentions

Students will:

- develop own definition of the term artificial intelligence (AI) and study its relationship with machine learning. With examples, they explore problems experienced in our society that can be solved by AI
- gain an insight into the mechanics of machine learning by training a simple model, then observe how AI is achieved by applying the model to a range of examples
- compare AI and machine learning with traditional programming to explain the difference between the two methods.

Preparation for the lesson

- Create a worksheet that allows students to:
 - state definitions of the words 'artificial' and 'intelligence'/'intelligent'
 - list the actions of a human, animal and machine that make them intelligent
 - describe their interpretation of the term 'artificial intelligence'. This can be derived from their answers to the previous questions.
- Develop a presentation with AI using supporting notes. Presentation should include a formal definition of AI. Source a video to give a broader definition and explanation of AI, such as:
 - HubSpotMarketing – What is Artificial Intelligence (or Machine Learning)?
<https://www.youtube.com/watch?app=desktop&v=mJeNghZXtMo&feature=youtu.be>.

- Prepare an incomplete table as per the example below. The type of tasks entered can be customised for the cohort.

Task	Example app	Human-like thinking or behaviour
Play games such as chess		
Navigation		Looking at a road map and deciding the best route to a destination based on several factors.
Advice		
Enhancing photos	Snapchat	
Translate languages		
Recommendation systems		Predicting popular movies based on previous selections.
Medical diagnosis		
Automation		

Table 1: Sample table for students to complete

- A collaborative spreadsheet with column headings can be created in an app such as Microsoft Office 365® with the link made available for students who are ready to contribute their answers.
- From the link below, collect and use examples of copyright-free images to train a model, such as orange versus apple, dog versus cat. The videos on the home page are a useful starting point.
 - Teachable Machine – What is Teachable Machine?
<https://teachablemachine.withgoogle.com/>.
- Create a worksheet with a table to list examples, human interpretation ('unsure', 'possibly' and 'certain' categories) and machine interpretation for students to record observations.

Support notes

- Artificial intelligence (AI) can be defined as machines that perform thinking and tasks associated with human intelligence. There is no single agreed definition of AI as the concept of intelligence is subjective.
- Recent advances in AI are a result of the implementation of machine learning.
- The idea of machine learning is based on the belief that data and pattern recognition is used to make decisions with a small amount of human intervention. The machine is trained to perform a specific task in a way that is similar to the human brain.
- A model is the output of a machine learning algorithm run on data. The model is made up of the model data and prediction algorithm.
- Supervised learning is providing a machine with examples, then programming the machine to learn from the examples.
- Reinforcement learning is providing a machine with feedback, then programming the machine to learn from the feedback.
- Programming machines to learn from experience removes the need for overly detailed programming.

- In traditional computer programming, a complex task is broken down into a set of instructions. These instructions are programmed into the machine to perform a task that is fixed at that time of programming.

Teaching and learning experience

Introduction

- Show the preselected video to expand on the definition of AI. Students must write three interesting points they discovered while watching the video.

Main activities

Activity 1

- Lead a discussion about what students found interesting.
- Provide students with an incomplete table and ask them to fill in the blank cells with an appropriate answer for applications of AI.
- On completion, students can add their ideas to a collaborative spreadsheet, trying not to double-up with data entered.

Activity 2

- Present an activity to train a computer to tell the difference between two similar objects. Train a model using Google – Teachable Machine (see previous link).
- As a class, watch the videos on the home page before attempting this activity. After training the model, use the preview (input option ‘File’) to confirm the confidence of the model in predicting categories. Use examples to test the model, recording observations on the prepared worksheet.
- Students create a list of everything in their lives that is artificial via an online live polling or brainstorming website.
- Ask students to complete the prepared worksheet, ideally using their own words and prior knowledge. On completion, students can swap their work with a partner (self-directed or lucky dip) then discuss feedback for one minute.
- Gather a consensus of opinions to create a class definition for AI. This could be in the form of a word cloud generator (online) or mind map (offline) for display.

Lesson conclusion

- Ask students to write the steps taken to train their model in Activity 2. They will now consider the differences between AI and machine learning and traditional programming. Organise the students to work in pairs to complete the prepared T-chart, listing differences in each method.

Lesson 2: Ethical considerations

The Western Australian Curriculum content addressed in this lesson is below.

Design thinking skills

Investigating and defining

- Investigate a range of technologies, resources and/or components to develop ideas and solutions, with consideration of social, ethical and other constraints

Learning outline

Learning intentions

Students will:

- explain the impacts of technology on society, with a focus on ethical considerations
- analyse different scenarios that stimulate personal values and judgements.

Preparation for the lesson

- Source sticky notes, pens and pencils.
- Create a presentation using the support notes and some examples of AI that promote ethical questions.
- Source a video that further explains an ethical issue related to AI. For example:
 - TED-Ed – The ethical dilemma of self-driving cars – Patrick Lin
<https://youtu.be/ixloDYVfKA0>.
- Create a *Six Thinking Hats* graphic organiser template.
 - The White Hat focuses on facts and data, asking what information is known and what is missing.
 - The Red Hat allows for emotional input, highlighting intuition without the need for justification.
 - The Black Hat takes a cautious, critical view identifying potential risks, problems or flaws.
 - The Yellow Hat emphasises optimism and benefits.
 - The Green Hat encourages creative thinking, seeking new ideas, possibilities and alternatives.
 - The Blue Hat is used for managing the thinking process, setting objectives and organising discussion.
- Create an online poll in a web app such as MS Forms. The poll will ask the user to select from a range of predetermined outcomes in the event of a crash involving an autonomous vehicle.
- Options for outcomes could include:
 - saves the greatest number of people
 - prioritises driver and passengers
 - is not programmed for this kind of event.

Support notes

- Keywords: ethics, abstraction, values, bias.

- The *Six Thinking Hats* is a creative problem-solving and decision-making framework developed by Edward de Bono. It helps individuals and teams approach issues from six perspectives, each represented by a coloured hat.
- Ethical considerations are a consequence of new or developing technologies such as those related to AI.
- The staged introduction towards fully autonomous vehicles has allowed for discussion of ethical decision-making.
- The introduction of machines that can think and make decisions has led to a range of questions, depending on the nature of the technology, including:
 - Will this mean loss of jobs?
 - Is a machine capable of being trusted?
 - Will we lose control?
 - What do we really mean by calling a machine intelligent?
 - Who is responsible for the decisions (owner, user, programmer, government)?
 - How will decisions be explained?
 - What about bias?

Teaching and learning experience

Introduction

- Give each student two sticky notes and ensure they have a pen or pencil. On one sticky note ask them to draw a pig, on the other a dog, both drawings without labels. Direct students not to show anyone their drawings.
- When finished, ask the students to swap their drawings with a partner and try to guess which drawing is the dog and which is the pig. Ask the students to consider their decision-making, and to give specific reasons for their guesses (e.g. the pig had a curly tail). However, this process is more complex than first thought as some dogs have curly tails).
- Finally, ask the students to reflect on the machine learning process from the previous lesson. Reinforce the level of complexity of algorithms needed to train a model, given the dog and pig activity, within the limits of confidence.

Main activities

Activity 1

- Present the main points from the support notes section.
- Play the sourced video that further explains an ethical issue related to AI as a case study of an application of AI to prompt discussion and questions. For example:
 - TED-Ed – The ethical dilemma of self-driving cars – Patrick Lin
<https://youtu.be/ixIoDYVfKA0>.

Activity 2

- Work through the scenarios at this link:
 - Moral Machine – Browse Scenarios
<https://www.moralmachine.net/>.
- If time allows, reinforce with the information at this link:
 - TED-Ed – The ethical dilemma of self-driving cars – Patrick Lin
<https://youtu.be/ixIoDYVfKA0>.

- Students use the *Six Thinking Hats* strategy to discuss the pros and cons of autonomous vehicles.
- In groups of six, each person takes on the role of one of the *Six Thinking Hats*, reflecting on their responses to the Moral Machine scenarios and the factors that influenced their decisions, such as:
 - saving more lives
 - gender preference
 - protecting law-abiding people
 - age preference
 - who is responsible in the event of an accident?

Lesson conclusion

- Ask students to place a vote in the prepared online poll.
 - Once voting is complete, use the graphical tools in the application to illustrate the results and present the results to the students.
 - Ask for individual feedback from students in an informal discussion.
 - Prompt the students with questions such as, 'Are these results surprising?'

Lesson 3: *Privacy Act 1988*

The Western Australian Curriculum content addressed in this lesson is below.

Privacy and security

- Australian Privacy Principles (APP) regarding the collection and ownership of data

Lesson outline

Learning intentions

Students will:

- develop familiarity with the *Privacy Act 1988* and examine the 13 Australian Privacy Principles that act as guidelines to support this law.

Preparation for the lesson

- Prepare a presentation including a video that demonstrates a relevant, ethical consideration for students, such as:
 - Sky News – Shock and anger as A-level algorithm strips pupils of their grades
<https://www.youtube.com/watch?app=desktop&v=rheYjX5exH0&feature=youtu.be>.
- Create a template of a Privacy spectrum, similar to Figure 1, to use as a student worksheet.
 Note: students might be reluctant to share their views given the sensitive nature of this activity.

PRIVACY SPECTRUM

Add examples of personal data, placing each in the appropriate category to match your values:

PRIVATE	CLOSE FRIENDS	FAMILY	COMPANIES	PUBLIC
100%				0%

Figure 1: Privacy spectrum

- Create worksheets to describe scenarios (including questions) based on the following fictitious societies. The worksheets should also ask for benefits and issues that could be encountered in each scenario.
 - Scenario 1: This society offers a complete right to privacy. Consider the possible reasons to take away a citizen's privacy.
 - Scenario 2: This society offers no right to privacy. Under what circumstances would there be a need for a citizen to be given privacy?

- Prepare a presentation that explains the *Privacy Act 1988* and summarises the 13 Australian Privacy Principles, as stated in the support notes.
- Create a reflective learning rubric for this activity.

Support notes

- Keywords: privacy, personal data, law, rights, responsibilities, values, cookies.
- Personal data is potentially shared online every time a user connects. Types of data include purchases, downloads, mouse movements and conversations.
- In Australia, the *Privacy Act 1988* protects the privacy of individuals and regulates the handling of personal information. This law is overseen by 13 Australian Privacy Principles which allow organisations to customise their procedures for handling of personal data and adapt their practices to ensure they follow the guidelines.
- Advertising and data collection organisations can place a cookie on a user's computer to match the type of advertisements to the user's interests, so they only see advertisements of interest while browsing. The interests are identified by the data stored in the cookie, which is a small data file stored in the browser to track website history and user activity. While data stored in cookies may not be of a sensitive nature, and therefore need not comply with the rules of the *Privacy Act 1988*, newer technology may integrate information given online.
- Terms and conditions for signing up to websites are offered at registration, however they are invariably lengthy, leading users to avoid taking the time to read them.
- Websites must disclose a privacy policy to state how personal data will be used and collected, as per the Australian Privacy Principles.
- The right to privacy must be balanced by the need for accountability in circumstances such as criminal investigations, for example, the use of closed-circuit television (CCTV) as evidence in a crime.

Teaching and learning experience

Introduction

- Use <https://www.moralmachine.net/> from the previous lesson to display three to five scenarios with judgements. Let students vote and then justify their selection.

Main activities

Activity 1

- Show the prepared video to demonstrate an ethical issue for students where bias is introduced. After viewing, allow sufficient time for students to do a *Think-pair-share* activity. The teacher can lead further discussion by asking students to explain their decisions.

Activity 2

- Distribute the scenario worksheets to students.
- Explain that ethics can extend to the use of personal data on the internet.
- Inform students about the *Privacy Act 1988* and how it is used to control the ways data is managed online.
- Ask students to visit the website below and make notes about the Rights and responsibilities.
 - Australian Government: Office of the Australian Information Commissioner: The Privacy Act <https://www.oaic.gov.au/privacy/the-privacy-act/>.

- Ask students to make notes about the 13 Australian Privacy Principles.
 - Australian Government: Office of the Australian Information Commissioner
<https://www.oaic.gov.au/privacy/australian-privacy-principles/australian-privacy-principles-quick-reference/>.
- Consider asking students to use Microsoft OneNote® for this task.
- Ask students to use their notes to create 10 questions and answers about the *Privacy Act 1988* that could be used to test a peer.
- Encourage students to submit their questions and answers to the teacher for reflective learning feedback.
Note: These questions will be used in the summative assessment (Appendix B).
- Ask students to complete the Privacy Spectrum worksheet by adding examples of personal data to the category they feel is correct. They may then add a statement to summarise their views about online privacy.
- In small groups, students choose one of the society scenarios. Each group completes the corresponding worksheet and shares their responses with the class.

Lesson conclusion

- Ask students to consider what constitutes their ideal society, in terms of privacy. Conduct a timed debate to challenge the ideal level of privacy in society.

Lesson 4: Chatbots

The Western Australian Curriculum content addressed in this lesson is below.

Digital systems

- Role of hardware and software to manage, control and secure the movement of data in a digital system

Lesson outline

Learning intentions

Students will:

- define the term 'chatbot' and explain its function in terms of AI
- examine several chatbots of differing levels of intelligence to comprehend the evolution of this technology
- carry out a Turing-style test and reflect on their interpretation of intelligence in machines
- consider an application of chatbots as virtual assistants for commercial websites. They will be critical of the potential for privacy concerns in the way this technology handles personal information.

Preparation for the lesson

- Create a slide presentation that summarises important points focusing on privacy concerns, prompted by the terms and conditions of a selected app.
- Choose an app that is popular with students and look at the terms and conditions. For example, when creating a Snapchat account, it can access your contacts, camera and microphone. Snapchat also learns all the other apps you have installed on your phone and information about your device, including the make and model of your phone and how much battery power remains.
- Create a presentation to reiterate an explanation of virtual assistants – Natural Language Processing (NLP) is recommended. Create this using chatbots, the support notes and the audiovisual link below.
 - NBC News – This Is The Algorithm That Lets Siri Understand Your Questions
https://youtu.be/uE_WJTngUwA.
- Using a website similar to the site linked below, demonstrate skills in training a chatbot to classify statements.
 - My computer brain – Artificial Intelligence Introduction: Anti-bullying AI
<https://mycomputerbrain.net/php/experiments/ai.experiment19b.php>.
- This activity can be related to Instagram's online bullying commitment as an example of a positive application of this technology.
 - Instagram – Our Commitment to Lead the Fight Against Online Bullying
<https://about.instagram.com/blog/announcements/instagrams-commitment-to-lead-fight-against-online-bullying>.
- Create a worksheet for a Turing-style test as per Figure 2.

Turing-style test

Chatbot: _____

Student: _____

Question	Response
What is the name of Bart Simpson's baby sister?	
What do you think of Roald Dahl?	
Are you a computer?	
What is the next number in the sequence 3, 6, 9, 12, 15?	
What do you think of nuclear weapons?	
What is 2×78 ?	
Do you like school?	

Did this chatbot seem to be an expert in one specific area only?

Were there any questions answered that did not make sense? Give an example.

Did the answers given appear to come from the same person? Why or why not?

Do you think this chatbot would pass as a human? Explain your answer.

Challenges:

- Try being a 'parrot' that simply copies what the chatbot says
- If you repeat an earlier question, do you get the same answer?
- Answer a question with a response that is clearly wrong
- Answer questions with responses from another chatbot

Figure 2: Turing-style test

Support notes

- A chatbot is a computer program that can have a conversation with a human and sound intelligent to varying degrees.
- Chatbots are programmed using traditional programming methods. The rules or algorithms that determine their responses are set at the time of programming. This means that the responses can never be changed unless they are reprogrammed. Refer to this chatbot:
 - Tell A.L.I.C.E.
<https://www.pandorabots.com/pandora/talk?botid=b8d616e35e36e881>.

- Refer to other chatbots (such as at the link below) to learn the rules of how to respond by analysing large databases of real conversations. The lack of consistent personality and inability to deal with new topics limits the intelligence of Cleverbot™.
 - Cleverbot
<http://www.cleverbot.com/>.
- The Turing test measures how well a computer can act like a human. To conduct the test, a human interrogator asks questions to a computer and a human, both located in separate rooms, via an instant messenger style program. The interrogator cannot identify where the responses are coming from. If the interrogator cannot tell the difference between the human and the computer responses, the computer is said to have passed the Turing test. Notably, a computer has never fully passed the Turing test.

Teaching and learning experience

Introduction

- Ask the students to consider the scenarios shown in the slide show. Allow sufficient time for students to conduct a *Think-pair-share* activity with a partner.
- Call on students to demonstrate, by standing or sitting, if they agree or disagree about the ethical use of apps such as Snapchat.
- Lead the discussion by asking students to explain their decisions. Do any students have difficulty in making a decision?

Main activities

Activity 1

- Lead a discussion on chatbots using the prepared presentation, video and applications.
- Students to complete the short activity (provided by the teacher). This demonstrates how to train AI to distinguish between kind versus mean comments. The activity shows what happens when a model is trained using different phrases. Explain to students this is based on how the human brain and its neural networks can learn.

Activity 2

- Students perform a Turing-style test to compare several chatbots with differing levels of AI; for example, Eliza, Alice and Cleverbot. Allocate a chatbot to the students and then have the class take part in a Jigsaw strategy.
- After the activity, students will join others who explored the same chatbot. Discuss the answers, comparing the level of intelligence shown in the responses of each chatbot.

Lesson conclusion

- Ask students to read the Privacy Policy for the Department of Social Services. Lead a discussion about the main points in Section 2.11 Electronic communication. Students can use the Australian Government, Department of Social Services – Privacy Policy <https://www.dss.gov.au/privacy-policy> as a guide.
- Ask students to position themselves in a Value line to indicate their level of confidence in personal information that is handled by AI. Ask students to give feedback on their choices and opinions.

Lesson 5: Review of algorithms, flow charts and programming structures

The Western Australian Curriculum content addressed in this lesson is below.

Digital implementation

- Design algorithms that use functions and represent them as flow charts and/or pseudocode

Design thinking skills

Designing

- Design alternative solutions considering available technologies, usability and aesthetics, using appropriate technical terms

Lesson outline

Learning intentions

Students will:

- review and expand on the definition of algorithms to include forms of iteration
- examine the three forms of iteration and practise skills in representing simple scenarios that require the use of iteration.

Preparation for the lesson

- Students may require a mathematical aid to draw flow chart shapes.
- Create a simple flow chart that represents sequence and selection. Leave the flow chart blank so students can contribute their ideas to complete the instructions.
- The algorithm could represent a coin toss guessing game, as shown in Figure 3.

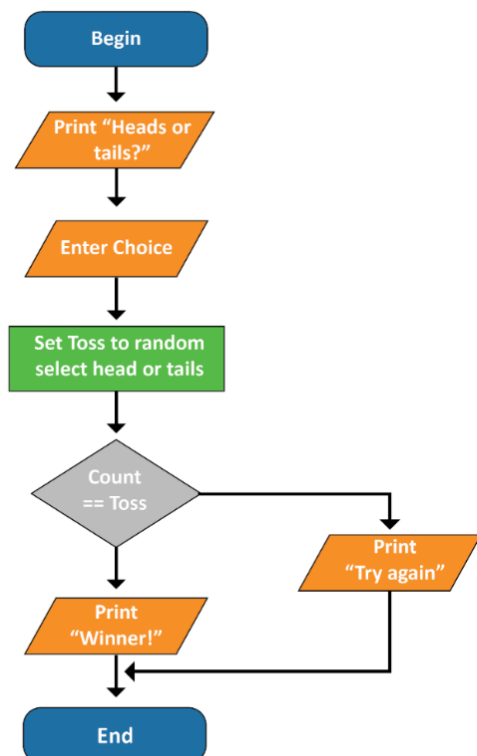


Figure 3: Flow chart illustrating coin toss guessing game

- Create a presentation that reviews past learning content for algorithms, then moves on to programming structures, with a focus on iteration.
- Create a worksheet with scenarios that reflect the three forms of iteration. Students will need to represent algorithms for each scenario as flow charts and structured English. For example, print the 13 Australian Privacy Principles (FOR); a number guessing game (WHILE); printing the product of two entered numbers that does not exceed 100 (REPEAT... UNTIL).

Support notes

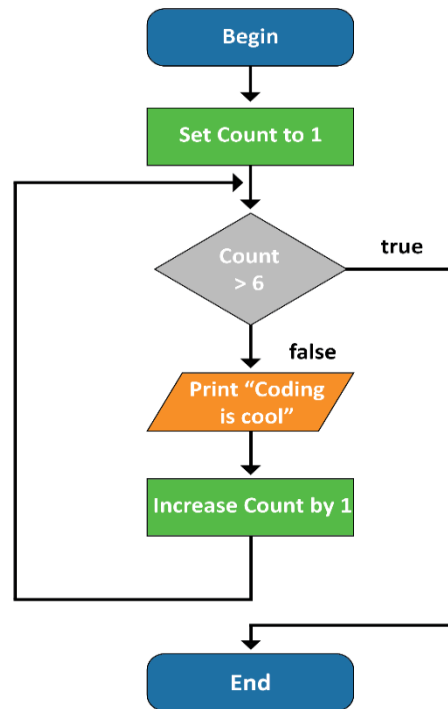
- An algorithm is a series of steps and decisions required to solve a problem. The steps must be completed in the correct order. Algorithms are essential in software development for the creation of efficient programs that are free of bugs (programming errors).
- Algorithms can be represented by structured English or visually, using flow charts.
- Structured English uses the English language to describe the steps in an algorithm in clear and precise statements. It can be a stepping stone to coding, especially when paired with keywords such as IF, THEN and UNTIL.
- Flow charts use symbols to represent structures, and arrows to identify the flow of logic.
- There are three programming structures that can be used as building blocks to represent any algorithm:
 - sequence is the ordering of instructions in a way that will produce the correct result
 - selection decides which instructions will be followed based on criteria
 - iteration is the repetition of instructions in an algorithm.
- Iteration is also called looping or repetition. There are three types of iteration, as shown in Figure 4.

FOR loops

Using the FOR command for a fixed (definite) number of iterations. The number of repetitions is known before entering the loop.

For example:

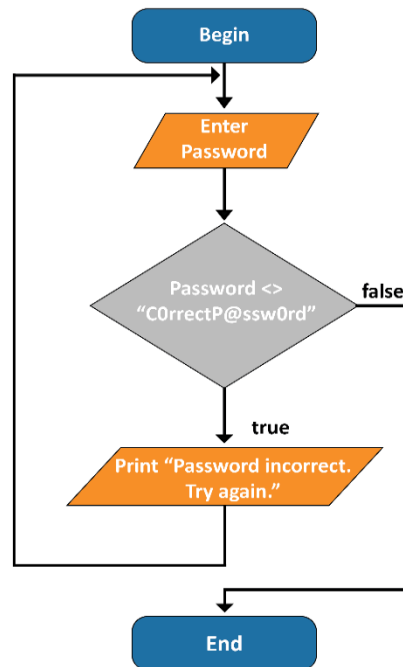
```
FOR Count ← 1 TO 6 do
  Output "Coding is cool"
EndFOR
```

**WHILE loops**

A WHILE loop runs a variable (indefinite) number of times. The number of repetitions is not known before entering the loop. The condition is pre-tested (test first), giving rise to the possibility that the loop might not be entered at all (zero iterations).

For example:

```
Input(Password)
While Password <>
  "C0rrectP@ssw0rd"
  Output "Password incorrect. Try again."
  Input(Password)
EndWhile
```



REPEAT loop (REPEAT UNTIL)

The REPEAT... UNTIL loop also completes a variable (indefinite) number of iterations, however the exact number is not known when the loop begins. This is tested at the end of the loop, (test last hence the loop will execute at least once).

For example:

```

Count ← 0
Repeat
  Output "Coding is cool"
  Count ← Count + 1
Until (Count = 10)
  
```

Note that Python does not have an option for a REPEAT... UNTIL loop. Instead, a 'while True' loop is used with a condition at the end that, when true, executes a 'break' statement.

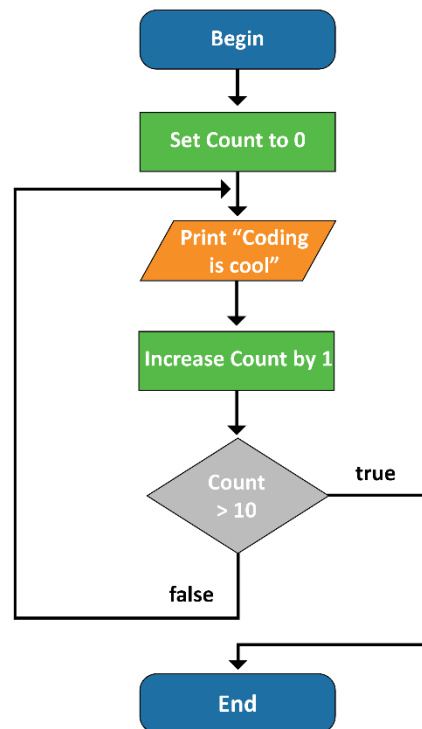


Figure 4: Flow charts for loops

Teaching and learning experience

Introduction

- Revise the topic of algorithms with students, in preparation for the Privacy chatbot assessment.
- Show students the prepared flow chart **without** instructions.
- Ask students to identify the shapes and their meaning.
- Give students the chosen problem and discuss the sequence of instructions needed to achieve the goal. They could do this by underlining the key requirements in a paragraph, then writing steps in structured English. Alternatively, steps could be printed on paper and cut out individually for students to place in order.
- Check with students that the sequence is correct, reiterating that steps not placed in order will lead to logic errors and a program that will not do what is required.
- Ask students for suggestions about how to transcribe steps into the flow chart, ensuring they write in the appropriate shapes, in order.

Main activities

Activity 1

- Tell students the flow chart created in the previous activity was correct, however ask them to imagine playing a game that closed, every time, on completion.
 - Ask students, Would this make it less fun to play than a game that continued to offer additional games on completion?
 - Request suggestions from students how they could add the option to play another game to the algorithm?
- Introduce the concept of iteration in a presentation that demonstrates the different ways iteration in algorithms can be implemented.
- Explain how iteration would solve the problem of providing the opportunity to play another game in the scenario given above. Which type of iteration would be needed?
- Students to complete the prepared worksheet to represent the algorithms for scenarios that reflect the three forms of iteration. Ask the students to submit their work at the end of the lesson for feedback prior to the next lesson.

Lesson conclusion

- Students write definitions for FOR, WHILE and REPEAT loops in their own words, and compare the three forms using a Venn diagram to show common and individual properties.

Lesson 6: Trace tables

The Western Australian Curriculum content addressed in this lesson is below.

Digital implementation

- Predict the output of an algorithm using a given range of test cases and compare against actual output

Design thinking skills

Designing

- Design alternative solutions considering available technologies, usability and aesthetics, using appropriate technical terms
-

Lesson outline

Learning intentions

Students will:

- validate the quality of algorithms through tracing techniques
- practise skills using trace tables.

Preparation for the lesson

- Source two examples of an algorithm that contains iteration; one will work correctly and the other will have an error in logic. Algorithms that do simple calculations are a good starting point, as any errors will be easy to find.
- Supply a worksheet that gives students the opportunity to complete trace tables. Use a range of algorithms in three sections, those that are:
 1. free of errors
 2. have errors that the students correct
 3. provide a final challenge that is slightly more complex.

The first trace table in each section should be partially complete to provide scaffolding for students who are unfamiliar with this task.

Support notes

- Keywords: trace tables, validation.
- Trace tables are used to check the logic of an algorithm before coding begins. Test data is used to test all pathways within an algorithm. All variables, constants and parameter values should be accounted for.

- A trace table has the line number of the program as rows, and elements such as variables and so on, as columns. See example below:

```

1   TotalScore <- 0
2   Continue <- "Y"
3   While Continue = "Y"
4     Input(Score)
5     TotalScore <- TotalScore + Score
6     Input(Continue)
7   EndWhile

```

Line	Total score	Continue	Continue="Y"	Score
1	0			
2		Y		
3			TRUE	
4				5
5	5			
6		Y		
3			TRUE	
4				8
5	13			
6		N		
3			FALSE	

Table 2: Trace table

Teaching and learning experience

Introduction

- Show a simple flow chart with a mathematical equation. For example, a number of variables divided by either 2, 4, 8, 16 or 24. Working in pairs, one student is allocated three different numbers to input and predict the output. Their partner will check if it is correct.

Main activities

- The teacher shows the first (correct) example of an algorithm that uses iteration, then explains the concept of a trace table by completing a trace table example. Repeat the activity for the students using the algorithm that contains a logic error.
- Ask students to complete the prepared worksheet on trace tables. Worksheets to be handed in at the end of the lesson for teacher feedback.

Lesson conclusion

- Students create their own algorithm as structured English that contains a logic error. This will be used as prior learning where students are required to trace and debug another student's algorithm at the start of Lesson 9.

Lessons 7–8: Revision of Python programming

The Western Australian Curriculum content addressed in these lessons is below.

Digital implementation

- Implement, modify and debug programs that use functions in a general-purpose programming language

Design thinking skills

Producing and implementing

- Select, implement and test a range of technologies, techniques and processes to produce designed solutions and/or prototypes

Lesson outline

Learning intentions

Students will:

- learn or revise Python programming skills to move on to subsequent lessons, including the assessment task (Appendix B).

Preparation for the lesson

- Source a Python program that demonstrates sequence, selection and iteration, such as a short, interactive game.
- Create a challenge for the students to solve using Python programming.

Support notes

- The following lessons will build on assumed prior knowledge of Python.
- On completion of Lessons 5 and 6, students will progress to practise representation of the programming structures in the Python programming language.

Teaching and learning experience

Introduction

- Show the class the Python code from the prior activities, then complete a code analysis with the students.
- Identify and explain the programming structures as well as variables and other significant elements. Trace data inputs and the intended outputs.

Main activities

- Provide students with a Python program to execute. The program will demonstrate sequence, selection and iteration, such as a short interactive game. Allow students to play the program several times.
- Discuss with the class how the program was created (input/output, processing, etc.). Identify a possible algorithm and make connections with the programming structures. Base the discussion on the observations the students made while running the code.

- Provide students with the program requirements that build on the structures in the Python program.
- Students will analyse the requirements and create algorithms (structured English and flow charts) for each mini-brief. They will use the algorithms to create Python programs as solutions to the briefs.
- Students solve the challenge (as provided by the teacher) using Python programming that is an extension on the previous activities. Students should use a structured English algorithm, a flow chart and a trace table to ensure their solutions are of a high quality.
- After completing the coding, students should work with at least two peers to test their program and provide feedback for improvement. For example, they can tell or create a random knock-knock joke, then ask the users if they wish to play again.

Lesson conclusion

- At the end of each activity, students will participate in the maintenance of a class wiki as a collaborative learning activity in an online community. They will communicate programming issues they encountered using structure and detail. Students to describe the solution that resolved the issue, if known.

Lesson 9: Python GUI

The Western Australian Curriculum content addressed in this lesson is below.

Digital implementation

- Design and prototype the user experience (UX) of a digital system based on user requirements

Design thinking skills

Designing

- Design alternative solutions considering available technologies, usability and aesthetics, using appropriate technical terms

Lesson outline

Learning intentions

Students will:

- analyse a variety of common graphical user interfaces (GUIs) to understand the functions and design common to most applications
- build a simple GUI using the tkinter module in Python.

Preparation for the lesson

- Create an online discussion board as a repository for the trace table algorithms from the conclusion of Lesson 6.
- Prepare a presentation on the definition of GUIs (see support notes).
- Provide students with a digital worksheet for a GUI analysis activity. This should include elements that make an efficient GUI. Students must be able to insert an image of their annotations, list and describe the elements as input/output, and explain functionality.
- Become familiar with creating simple GUIs using widgets from the Python tkinter module.
An excellent resource for tkinter is:
 - Teachers pay teachers – tkinter Tutor
<https://www.teacherspayteachers.com/Product/tkinter-Tutor-3708421>
- Become familiar with the PC DOS 5 operating system.
 - PCE.js IBM PC/XT emulator running PC DOS – a hack by James Friend
<https://jamesfriend.com.au/pce-js/ibmpc-games/>
 - Enter the following commands:
 - *mkdir myfiles*: create a folder called 'myfiles'
 - *cd myfiles*: open the folder 'myfiles'
 - *edit newfile.txt*: creates file 'newfile.txt' and opens the editor. Type some text, then press ALT to access the menus. Use hotkeys (shown in white) to navigate the menus.

Support notes

- Keywords: graphical user interface (GUI), event-driven, user-friendly, console, window, icon, menu, pointer.

- A GUI is a software application that enables the user to interact with the program. They are event-driven programs with visual icons and instructions. This alters the flow of control from that of a procedural program. With a GUI, the order in which code is executed is determined by events triggered by the user. The events are triggered by a variety of input methods, including mouse and voice-control.
- Modern computers are more intuitive and user-friendly than their predecessors, which is largely due to the graphical nature of user interaction.
- A well-designed GUI should consider:
 - targeted use of design concepts
 - ease of use
 - language suitable for the target audience
 - documentation, including help manuals.

Main activities

Activity 1

- Prepare a presentation about GUIs, based on the support notes, that will inform students of the purpose, composition, and design of GUIs.
- Demonstrate coding a simple interface, using tkinter in Python, with students observing.

Activity 2

- Have students analyse the GUI of a common application and a chatbot.
- Students to complete the prepared digital worksheet, taking a screenshot of the interface and describing the elements that make up the interface. Suitable apps could include Microsoft Word® and Cleverbot. Identify windows, icons and menus, as well as any elements unique to the application.
- Students then swap seats with a partner and evaluate their partner's work. All students then gather and take one minute to share each other's findings.

Lesson conclusion

- Students try the character-based screen using the online emulator resource (see Preparation for the lesson notes). Discuss with students the advantages and disadvantages of using the early, console-based (or character-based) computers compared with contemporary devices.

Lesson 10: Python CSV file import

The Western Australian Curriculum content addressed in this lesson is below.

Digital implementation

- Implement, modify and debug programs that use functions in a general-purpose programming language

Design thinking skills

Producing and implementing

- Select, implement and test a range of technologies, techniques and processes to produce designed solutions and/or prototypes
-

Lesson outline

Learning intentions

Students will:

- understand the structure of a CSV file and how it can be used to act as a dynamic transfer medium for data
- gain practical skills in importing data from a CSV file into a Python program.

Preparation for the lesson

- Create an online quiz based on concepts learned in previous lessons about GUIs.
- Prepare two CSV files for import into a Python program that will display the data in a common format. For example, a CSV file containing column A with numbers 1 to 13 and column B a short description of the Australian Privacy Principle number 11. A second CSV file might be a list of computer jokes, with column A the question and column B the answer.
- Students may need mathematical aids to complete flow charts, or to produce them electronically in Word. Create graphic organisers for trace tables and testing documents, or prepare for students to create, where required.
- Preselect CSV file options for students from one of these websites, as appropriate.
 - Australian Government, Data.gov.au – Search for data
<https://www.data.gov.au/search?format=csv>.
 - Digital Technologies Hub – Data Science STEM resources
<https://www.digitaltechnologieshub.edu.au/teachers/lesson-ideas/integrating-digital-technologies/data-science-stem-resources>.
- Create a collaborative PowerPoint® in Office 365, with title page only, and share an editable link with students.

Support notes

- Keywords: Comma-separated values (CSV), delimiter, import, export.
- A comma-separated values or CSV file is a flat file containing data that can be imported into a Python program. Data can also be exported to create a CSV file using Python code.
- CSV files are plain text files that contain only printable characters.
- The file stores data in a tabular format using commas to separate fields (values). See example in Figure 5. Other characters can be used to separate fields (e.g. tab, semicolon); these character separators are called delimiters. Note: if the delimiter is a comma, then data should not contain commas, or an alternative delimiter used.

```
Name,Subject,Mark,Grade
John,Science,85,A
Jane,English,50,C
Joyce,Maths,36,D
```

Figure 5: CSV data example

- A new line of the CSV file represents a new row (record).
- It is advantageous to import a CSV file rather than hard coding data within a program. Data that is stored in a program cannot be changed unless the code is reprogrammed. Programming code is also less complex when data is stored externally, making the maintenance of the code easier and reducing the likelihood of errors.
- CSV files can be used to import data into a Python program as it is executed. They are easily edited in Notepad or Microsoft Excel®, as appropriate.
- CSV files containing data for differing purposes can reuse the same algorithm, and therefore the same program, for processing.

Formative assessment

- The following checklist can be used as a guide for the formative assessment.

Above satisfactory	Satisfactory	Below satisfactory (needs work)
Flow chart uses correct symbols for process, arrows for logic flow and represents the algorithm with accuracy.	Flow chart uses mostly correct symbols for process, arrows for logic flow and represents the algorithm with reasonable accuracy.	Flow chart uses few correct symbols for process, missing arrows for logic flow and is not representative of algorithm.
Structured English (algorithms) also represents the algorithm in completion and with accuracy. Transition to program code is seamless.	Structured English (algorithms) is mostly representative of the algorithm. Transition to program code is straightforward.	Structured English (algorithms) is incomplete and with errors.
Trace tables show consistently accurate evidence to predict quality of code.	Trace tables show basic evidence of quality of code.	Trace tables not completed or incorrect.

Above satisfactory	Satisfactory	Below satisfactory (needs work)
Python program reflects algorithm with least complexity. Program is commented to provide information about all code blocks.	Python program is reflective of algorithm with attention to limiting complexity. Program has some comments to inform of code blocks	Python program reflects lack of understanding of algorithm. It may not run and does not contain comments.
Evidence of user testing is comprehensive, structured, and well-documented. Feedback for improvement demonstrates action taken and reflection.	Basic evidence of user testing is documented with feedback for improvement acknowledged with simple actions.	Evidence of user testing is minimal.

Table 3: Formative assessment guide

Teaching and learning experience

Introduction

- Students participate in an online quiz to review previous learning about GUIs.

Main activities

Activity 1

- Demonstrate how to import two CSV files using the same Python program. Students propose ideas to explain how two separate data files could be compatible with the same Python program.
- Extend the conversation to include the pros and cons of using CSV files compared to storing data in a program.
- See Figure 6 for an example of a Python CSV import program.

```
# Python's built-in CSV library
import csv

# Initialise dictionary variable
dict_from_csv = {}

# Open the CSV file
with open('csv_file.csv', encoding="utf-8-sig") as inp:
    # Read in the data
    reader = csv.reader(inp)
    # Create dictionary
    for row in reader:
        dict_from_csv[row[0]] = row[1]

# Print dictionary keys and values
for q,a in dict_from_csv.items():
    print(q,a)
```

Figure 6: Python CSV import program example

Activity 2

- Create a CSV file and a Python import program for a sample scenario, such as a list of online gamer names and their scores.
- Extend the challenge by asking students to enter a player name and display that player's score.
- The teacher may collect this activity for a formative assessment. Students can also provide evidence of planning, such as flow charts, structured English, trace tables and testing.

Lesson conclusion

- Students examine real-life datasets by downloading CSV files from one of the following websites:
 - Australian Government: Data.gov.au – Search for data
<https://www.data.gov.au/search?format=csv>.
 - Digital Technologies Hub – Data Science STEM resources
<https://www.digitaltechnologieshub.edu.au/teachers/lesson-ideas/integrating-digital-technologies/data-science-stem-resources>.
- Students select one CSV file and:
 - state the context of the data collection (e.g. background information, source)
 - describe the data included in the file using column headings and scanning data
 - complete a simple analysis of the data, taking into account the context, using sort and filter tools on columns of interest
 - create three questions that could be answered by the data.
- Students share their responses by contributing one slide to a class PowerPoint in Office 365.

Lessons 11–15: Python privacy chatbot

The Western Australian Curriculum content addressed in these lessons is below.

Digital systems

- Role of hardware and software to manage, control and secure the movement of data in a digital system

Privacy and security

- Australian Privacy Principles (APP) regarding the collection and ownership of data

Digital implementation

- Design algorithms that use functions and represent them as flow charts and/or pseudocode
- Predict the output of an algorithm using a given range of test cases and compare against actual output
- Implement, modify and debug programs that use functions in a general-purpose programming language

Design thinking skills

Investigating and defining

- Investigate a range of technologies, resources and/or components to develop ideas and solutions, with consideration of social, ethical and other constraints

Designing

- Design alternative solutions considering available technologies, usability and aesthetics, using appropriate technical terms

Producing and implementing

- Select, implement and test a range of technologies, techniques and processes to produce designed solutions and/or prototypes

Evaluating

- Evaluate design processes and solutions against student-developed criteria

Lesson outline

Learning intentions

Students will:

- complete the Python privacy chatbot sample task to the best of their ability, demonstrating the content studied in previous lessons, using creativity and skill.

Preparation for the lesson

- Create a worksheet for the evaluation where students take notes while evaluating each other's chatbot. Questions to consider including on the worksheet to relate to:
 - relevance of privacy quiz questions
 - sketch of GUI, including annotations
 - ease of use
 - design
 - appeals to target audience
 - presence or absence of bugs (programming errors)
 - how chatbot has improved the player's knowledge of the *Privacy Act 1988*.

Support notes

- Familiarise yourself with the Python privacy chatbot task (Appendix B).
- Check that students are familiar with the content and skills required to complete the task to the best of their ability.
- In addition to Python, students should choose the software tools they are most familiar with to complete the task.
- Consider differentiating the task for lower ability students. The tkinter GUI can be replaced with simple output to the Python console by using the Print() function.

Teaching and learning experience

Introduction

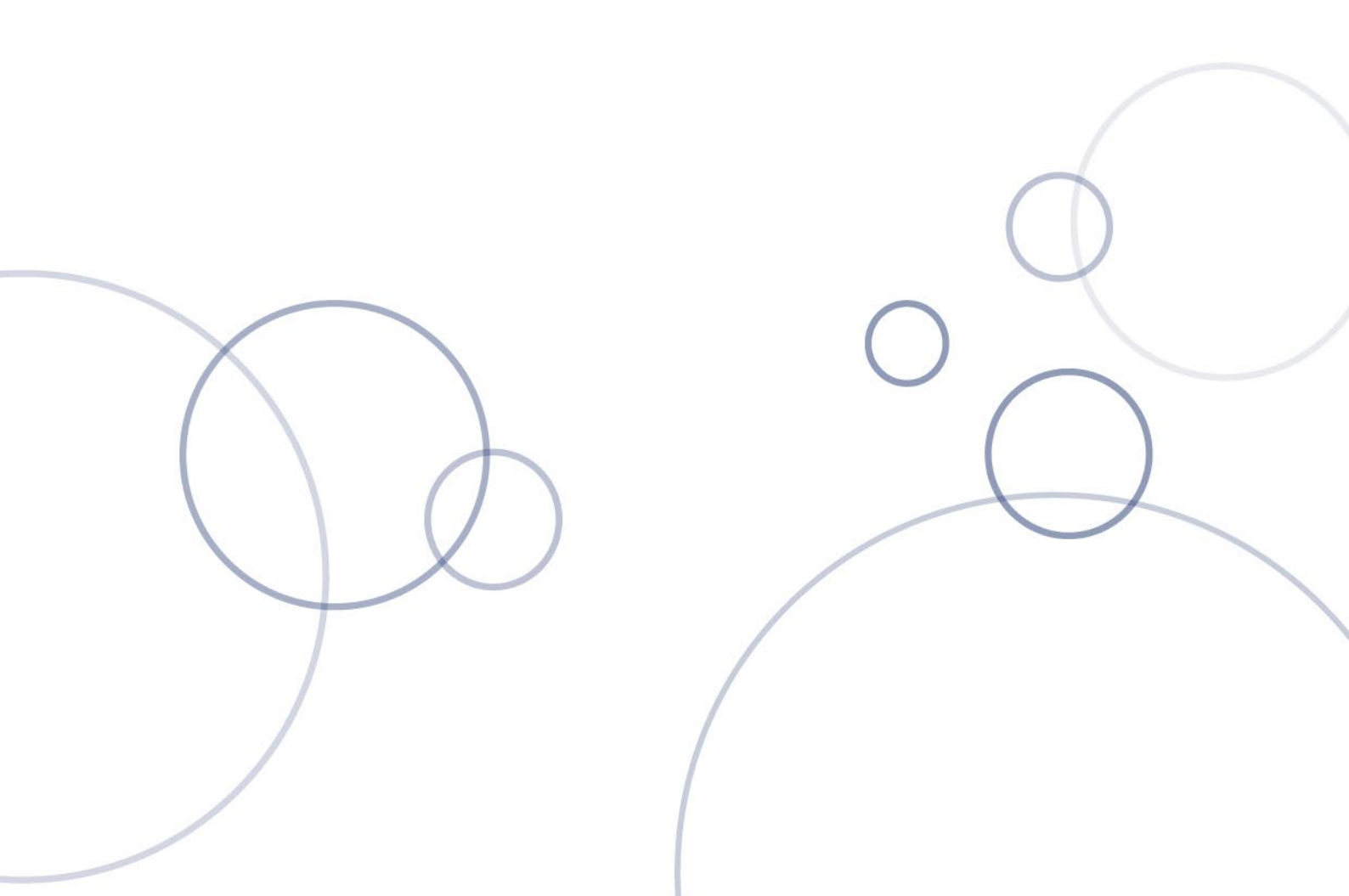
- Students recap their knowledge by creating a mind map of their learning journey. They could use pen and paper with colour or a web app such as:
 - bubbl.us – Mind Mapping Online
<https://bubbl.us/>.

Main activities

- Explain each element of the task relevant to the individual requirements and expectations of each of the lessons.
- This task can be broken down into sections, with milestones for student planning and feedback. Exemplars of student work may assist with the explanation.
- Complete the Python privacy chatbot task (Appendix B).

Lesson conclusion

- On completion of this task, students evaluate each other's chatbot, and complete the evaluation worksheet, gathering information to determine their own success criteria for programming a chatbot.



Appendix A

Resources

Appendix A: Resources

Lesson	Resource	Link/information
1	HubSpot Marketing – What is Artificial Intelligence (or Machine Learning)?	https://www.youtube.com/watch?app=desktop&v=mJeNghZXtMo&feature=youtu.be
	Teachable Machine – What is Teachable Machine?	https://teachablemachine.withgoogle.com/
2	TED-Ed – The ethical dilemma of self-driving cars – Patrick Lin	https://youtu.be/ixloDYVfKA0
	Moral Machine – Browse Scenarios	https://www.moralmachine.net/
3	Sky News – Shock and anger as A-level algorithm strips pupils of their grades	https://www.youtube.com/watch?app=desktop&v=rheYjX5exH0&feature=youtu.be
	Australian Government, Office of the Australian Information Commissioner – <i>The Privacy Act</i>	https://www.oaic.gov.au/privacy/the-privacy-act/
	Australian Government, Office of the Australian Information Commissioner – Australian Privacy Principles quick reference	https://www.oaic.gov.au/privacy/australian-privacy-principles/australian-privacy-principles-quick-reference/
4	NBC News – This Is The Algorithm That Lets Siri Understand Your Questions	https://youtu.be/uE_WJTnqUwA
	My computer brain – Artificial Intelligence Introduction: Anti-bullying AI	https://mycomputerbrain.net/php/experiments/ai.experiment19b.php
	Instagram – Our Commitment to Lead the Fight Against Online Bullying	https://about.instagram.com/blog/announcements/instagrams-commitment-to-lead-fight-against-online-bullying
	Pandorabots – Tell A.L.I.C.E.	https://www.pandorabots.com/pandora/talk?botid=b8d616e35e36e881
	Cleverbot	http://www.cleverbot.com/
	Australian Government, Department of Social Services	https://www.dss.gov.au/privacy-policy

Lesson	Resource	Link/information
9	Teachers Pay Teachers – tkinter Tutor, tkinter for Beginners (Complete course)	https://www.teacherspayteachers.com/Product/tkinter-Tutor-3708421
	PCE.js IBM PC/XT emulator running PCDOS 5 – a hack by James Friend	https://jamesfriend.com.au/pce-js/ibmpc-games/
10	Australian Government, Data.gov.au – Search for data	https://www.data.gov.au/search?format=csv
	Digital Technologies Hub – Data Science STEM resources	https://www.digitaltechnologieshub.edu.au/teachers/lesson-ideas/integrating-digital-technologies/data-science-stem-resources

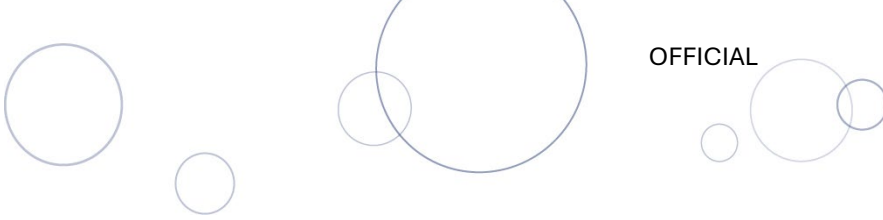
Self-evaluation template

The self-evaluation template below is a sample to aid teachers. In Year 9, students 'Evaluate design processes and solutions against student-developed criteria'. Students are to create their own criteria for self-evaluation.

Referring to the worksheet from the peer evaluation, list the criteria you determined for creating a successful chatbot.

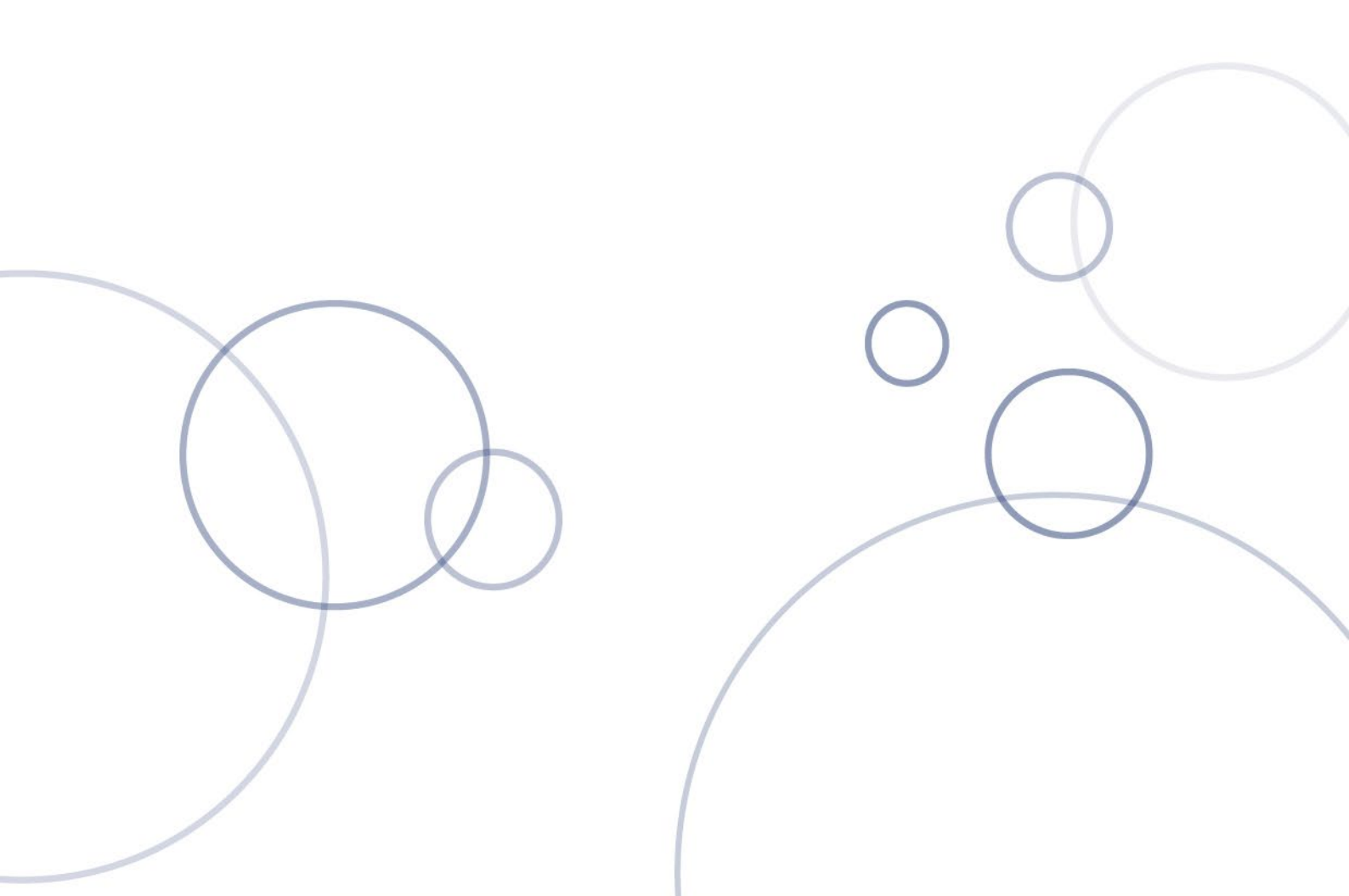
Describe how your chatbot met the success criteria listed above.

Describe how your chatbot did **not** meet the success criteria listed.



Explain the feedback other students have about your chatbot.

How could you change a future chatbot project to make sure that it meets all the listed criteria?



Appendix B

Summative assessment task

Python privacy chatbot

Task details

Title	Python privacy chatbot
Description	Students use the Python program to code a chatbot and demonstrate knowledge of digital implementation including the <i>Privacy Act 1988</i> .
Type of assessment	Summative
Purpose of assessment	To assess students' understanding of digital implementation including the <i>Privacy Act 1988</i> . To assess students' skills in designing and producing a digital product for the purpose of delivering digital implementation content.
Ways of assessing	Practical evidence, observations, graphic organiser
Evidence to be collected	Chatbot program Design, content and planning Evaluation and self-reflection
Suggested time	Five one-hour lessons
Differentiation	Teachers should differentiate their teaching and assessment to meet the specific learning needs of their students, based on their level of readiness to learn and their need to be challenged. Where appropriate, teachers may either scaffold or extend the scope of the assessment tasks.

Content descriptions

Digital systems

- Role of hardware and software to manage, control and secure the movement of data in a digital system

Privacy and security

- Australian Privacy Principles (APP) regarding the collection and ownership of data

Digital implementation

- Design algorithms that use functions and represent them as flow charts and/or pseudocode
- Predict the output of an algorithm using a given range of test cases and compare against actual output
- Implement, modify and debug programs that use functions in a general-purpose programming language

Design thinking skills

Investigating and defining

- Investigate a range of technologies, resources and/or components to develop ideas and solutions, with consideration of social, ethical and other constraints

Designing

- Design alternative solutions considering available technologies, usability and aesthetics, using appropriate technical terms

Producing and implementing

- Select, implement and test a range of technologies, techniques and processes to produce designed solutions and/or prototypes

Evaluating

- Evaluate design processes and solutions against student developed criteria

Task preparation

Prior learning

- Students have completed Lessons 1–10.
- Students are allocated class time to plan, prepare and produce their task.

Resources

- Python development software

Instructions to teacher

- Ask students to refer to the 10 questions and answers they created in Lesson 3 in this learning sequence and the reflective learning feedback.
- Provide students with the opportunity to act on the feedback provided.
- Students are to transfer the 10 questions and answers they created in Lesson 3 in this learning sequence into an appropriate CSV format.
- Ask students to complete a sketch of the GUI of their chatbot to show required input and output. Annotations should include widgets, triggers for events, proposed content and design features.
- Have students create planning documents such as flow charts, structured English and trace tables.
- Instruct students to code a chatbot using Python programming language, including a tkinter module, to create the GUI.
- Students must complete a detailed self-reflection in the form of a visual and/or written diary.
- On completion of the task, students share their final product to peers for evaluation.
- Using a worksheet to provide a scaffold, the peer evaluation will provide information which will form the basis of success criteria for self-evaluation.

Instructions to students

Task description: Python privacy chatbot

Name: _____

Working individually, create a Python privacy chatbot to demonstrate your understanding of the concepts related to the *Privacy Act 1988*, as previously studied. The chatbot will be in the form of a quiz, asking questions about the concept of privacy and displaying the correct answers. A graphical user interface (GUI) will be created to interact with the player. Questions and answers will be imported from a CSV file.

Part A: Investigating and defining

1. Explain the need for laws such as the *Privacy Act 1988*, especially in relation to the impacts of technology. (3 marks)

2. Describe artificial intelligence (AI). (2 marks)

3. Describe three ethical issues that have developed with the introduction of AI. (6 marks)

- i.

- ii.

- iii.

4. Describe the Turing test. How many computers have passed the Turing test? (3 marks)

5. Explain the process of training a computer to tell the difference between two images, such as a dog and a cat. (3 marks)

6. Outline the purpose of trace tables. (2 marks)

7. Explain the difference between a graphical user interface (GUI) and a console-based interface. (3 marks)

8. Identify the software application you chose. Explain your reasons for this selection. (4 marks)

9. List three characteristics of the target audience. (3 marks)

10. Describe how you will create your chatbot so that it appeals to the target audience. (2 marks)

11. Sketch the proposed GUI for the chatbot, with annotations to explain widgets, events, design features and proposed content. (4 marks)



Part B: Designing

Software design

1. Create structured English to represent the chatbot algorithm. (13 marks)

2. Create a flow chart to represent the algorithm for the chatbot and paste below. (13 marks)

3. Complete a trace table to verify your algorithm. (10 marks)

LINE NUM							

Part C: Producing and implementing, and Evaluating

Students create the Python privacy chatbot.

(18 marks)

Students evaluate the Python privacy chatbot. Refer to self-evaluation template (Appendix A).

(4 marks)

Marking key

Part A: Investigating and defining

Description	Marks
1. Explain the need for laws such as the <i>Privacy Act 1988</i>, especially in relation to the impacts of technology.	
Explains the <i>Privacy Act 1988</i> in relation to the impacts of technology	3
Describes the <i>Privacy Act 1988</i> in relation to the impacts of technology	2
Identifies the <i>Privacy Act 1988</i>	1
Subtotal	/3
Possible answers could include: The <i>Privacy Act 1988</i> aims to keep data safe and secure when users interact with business, government and organisations. The Act indicates how and why data can be used and stored in Australian states and territories. This Act plays an important role in reducing the negative impacts of technology on users and build user confidence in innovative technologies.	
2. Describe artificial intelligence (AI).	
Describes AI	2
Makes a general comment about AI	1
Subtotal	/2
Possible answers could include: Artificial Intelligence refers to a computer system or machine that demonstrates the ability to learn and mimic intelligence through the use of complex algorithms.	
3. Describe three ethical issues that have developed with the introduction of AI.	
Describes an ethical issue that has developed with the introduction of AI (two marks each)	2
Makes a general comment about ethical issues associated with AI	1
Subtotal	/6
Possible answers could include: - AI can cause unemployment for unskilled labour through automation. - AI technology can develop a bias, based on the data sets incorporated. - AI technology can infringe on the privacy of civilians if data is not secured and restricted by the systems.	
4. Describe the Turing test. How many computers have passed this test?	
Describes the Turing test	2
Makes a general comment about the Turing test	1
Identifies the number of computers that have passed the Turing test (one mark only)	1
Subtotal	/3
Possible answers could include: The Turing test refers to a machine/system's ability to replicate intelligence that is indistinguishable from humans. No computers have successfully passed the Turing test.	

Description	Marks
5. Explain the process of training a computer to tell the difference between two images, such as a dog and a cat.	
Explains the process of AI training to tell the difference between two images	3
Gives an outline of the process of AI training to tell the difference between two images	2
Provides superficial information about AI training to tell the difference between two images	1
Subtotal	/3
Possible answers could include: Training of AI starts with the collection of images, for example, dogs and cats, to which the outcome is clear. These need to be large data sets for accurate training. An algorithm is created to classify images. The AI is trained manually until the intended accuracy levels are achieved.	
6. Outline the purpose of trace tables.	
Outlines the purpose of trace tables	2
Makes superficial comments about trace tables	1
Subtotal	/2
Possible answers could include: A trace table is a tool created to desk check an algorithm. A trace table is created after planning and before programming. The purpose of the trace table is to match the intended outcome of the inputs (as recorded on paper) with the actual outputs of the program (once completed in the testing phase).	
7. Explain the difference between a graphical user interface (GUI) and a console-based interface.	
Explains in detail the difference between a GUI and a console-based interface	3
Gives an outline of the difference between a GUI and a console-based interface	2
Provides superficial information about the difference between a GUI and a console-based interface	1
Subtotal	/3
Possible answers could include: A GUI is a visual interface that represents how users interact with a system or software. GUIs involve graphics and objects that are arranged with useability in mind. Console-based interfaces are text-based.	
8. Identify the software application you chose. Explain your reasons for this selection.	
Identifies the correct software application (one mark only)	1
Explains in detail several reasons for this selection	3
Describes reasons for this selection	2
Identifies reasons for this selection	1
Subtotal	/4

Description	Marks
9. List three characteristics of the target audience.	
One mark per correct characteristic of the intended target audience	1–3
Subtotal	/3
Possible answers could include: Characteristics can include: • Age • Gender • Lifestyle • Interests	
10. Describe how you will create your chatbot so that it appeals to the target audience.	
Describes how to create a chatbot that is appealing to the target audience	2
Identifies ways to create a chatbot that is appealing to the target audience	1
Subtotal	/2
11. Sketch the proposed GUI for the chatbot, with annotations to explain widgets, events, design features and proposed content.	
Develops a detailed sketch design with complete and accurate annotations that explain widgets, events, design features and proposed content	4
Develops a sketch design with some detailed annotations. States the events and content	3
Develops a basic sketch design with some annotations and events	2
Develops part of a sketch	1
Subtotal	/4
Part A total	/35

Marking key

Part B: Designing

Description	Marks
1. Create structured English to represent the chatbot algorithm	
Algorithm: logic/sequence	
Creates logical sequence of steps with a high level of efficiency in the solution. Uses all the correct control structures, such as variables	5–6
Identifies sequence of steps for the solution. Uses correct control structures, such as variables	3–4
Attempts to create a sequence of steps with limited efficiency in the solution. Uses some control structures, such as variables	1–2
Subtotal	/6
Algorithm: selection/decision	
Consistently identifies one-way and two-way decisions (branching) correctly and implements efficiently	4
Consistently identifies one-way or two-way decisions (branching) correctly	3
Identifies one-way and/or two-way decisions (branching) correctly	2
Implements one-way and/or two-way decisions (branching) incompletely or incorrectly	1
Subtotal	/4
Algorithm: iteration	
Successfully includes iteration (repetition), where required, with efficiency and accuracy	3
Successfully includes iteration (repetition), where required	2
Unsuccessfully attempts to include iteration (repetition)	1
Subtotal	/3
2. Create a flow chart to represent the algorithm for the chatbot	
Flow chart: logic/sequence	
Provides a logical sequence of steps with a high level of efficiency in the solution. Correct control structures used, such as variables	5–6
Identifies a sequence of steps for the solution. Correct control structures used, such as variables	3–4
Attempts to include a sequence of steps with limited efficiency in the solution. Mostly correct control structures used, such as variables	1–2
Subtotal	/6

Description	Marks
Flowchart: selection/decision	
Consistently identifies one-way and/or two-way decisions (branching) correctly and implemented these efficiently	4
Consistently identifies one-way and/or two-way decisions (branching) correctly	3
Correctly identifies one-way and/or two-way decisions (branching)	2
Implements incomplete or incorrect one-way and/or two-way decisions (branching)	1
Subtotal	/4
Flowchart: iteration	
Includes iteration (repetition), where required with efficiency and accuracy	3
Includes iteration (repetition), where required	2
Some attempt to include iteration (repetition)	1
Subtotal	/3
3. Complete a trace table to verify your algorithm	
One mark for each correct line in the table	1–10
(This will vary depending on the algorithm created)	
Subtotal	/10
Part B total	/36

Marking key

Part C: Producing and implementing, and Evaluating

Description	Marks
Producing and implementing	
Creates and refines an effective digital solution that has an engaging user experience	7–8
Creates effective digital solutions that considers the user experience	5–6
Creates digital solutions that considers the user experience	3–4
Creates digital solutions with minimal consideration for the user experience and are incomplete or inappropriate	1–2
Subtotal	/8
Producing and implementing	
Uses a range of techniques, software and hardware safely, to create a chatbot experience that has accurate and detailed content	9–10
Uses a range of techniques, software and hardware safely, to create a chatbot experience that mostly has well-selected content	7–8
Uses some techniques, software and hardware safely, to create a chatbot experience that demonstrates appropriate content	5–6
Uses some techniques, software and hardware safely, to create part of a chatbot experience that demonstrates some appropriate content	3–4
Uses limited techniques, software and hardware safely, to create part of a chatbot experience that demonstrates limited content	1–2
Subtotal	/10
Evaluating	
Analyses a range of other students' work to fully develop success criteria for this product. Uses this success criteria with feedback received from others to complete a comprehensive self-evaluation, including a detailed reflection of all stages of the design process	4
Analyses several examples of other students' work to develop success criteria for this product. Uses this success criteria with feedback received from others to complete a self-evaluation, including a detailed reflection of all stages of the design process	3
Compares examples of other students' work to describe some success criteria for this product. Uses this success criteria with feedback received from others to complete a basic self-evaluation, with a focus on some stages of the design process	2
Self-evaluation is limited or incomplete	1
Subtotal	/4
Part C total	/22
Assessment total	/93

Acknowledgements

Lesson 2: Ethical considerations

Activity 3

Concept from: de Bono, E. C. F. P. (1985). *Six Thinking Hats*. Little Brown.

Appendix B

Marking key

Information from: Attorney-General's Department. (1988).

Privacy Act 1988.

<https://www.legislation.gov.au/C2004A03712/latest/text>

