Government of Western Australia
School Curriculum and Standards Authority

# Sample Teaching and Learning Outline

## Technologies

## Digital Technologies

## Year 9

**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 teaching and learning programs. Their inclusion does not imply that they are mandatory or that they are the only resources relevant to the learning area syllabus.

2020/21699v4

This document is an introduction to planning a teaching and learning outline with syllabus content for Year 9 Digital Technologies. It provides suggested sequencing and timing for teaching the syllabus content. For further details on curriculum requirements and available options, teachers should refer to the School Curriculum and Standards Authority's (the Authority's):

- *Policy* Standards *for Pre-primary to Year 10: Teaching, Assessing and Reporting*
- Table 1: *Western Australian Curriculum and Assessment Outline*: curriculum requirements and available options.

Schools may choose to teach the syllabus content for two hours per week for a semester, **or** one hour per week for the year. Sample plans provide a range of possible learning experiences from which assessments should be drawn. This *Year 9 Sample Teaching and Learning Outline* provides teachers with possible learning experiences over eight weeks and unpacks the syllabus content to assist teachers in their understanding.

A presentation (*Western Australian Curriculum Technologies Presentation*), which unpacks the process to develop this plan, is available on the Presentations page of the Authority website (https://k10outline.scsa.wa.edu.au/home/resources/presentations).

## Year 9 Syllabus Content – Digital Technologies

| Content | Description |
|---|---|
| **Digital systems** | Role of hardware and software in managing, controlling and securing the movement of data in a digital system |
| **Representation of data** | Different methods of manipulation, storage and transmission of data |
| **Collecting, managing and analysing data** | Explore techniques for acquiring, storing and validating quantitative and qualitative data<br>Analyse and visualise data to create information and address complex problems |
| **Digital implementation** | Design the user experience of a digital system<br>Design algorithms, represented diagrammatically and in structured English, and validate plans and programs through tracing<br>Implement and apply data storage and organisation techniques<br>Create and use interactive solutions for sharing ideas and information online, taking into account social contexts |
| **Investigating and defining** | Identify and define the needs of a stakeholder, to create a brief, for a solution<br>Investigate a selection of components/resources to develop solution ideas, identifying and considering constraints |
| **Designing** | Apply design thinking, creativity and enterprise skills (WATPPS56)<br>Design solutions assessing alternative designs against given criteria, using appropriate technical terms and technology |
| **Producing and implementing** | Select, and safely implement and test appropriate technologies and processes, to make solutions |
| **Evaluating** | Evaluate design processes and solutions against student developed criteria |
| **Collaborating and managing** | Work independently, and collaboratively to manage projects, using digital technology and an iterative and collaborative approach. Considers time, cost, risk and safety |

**Year Level Description**

In Year 9, learning in digital technologies focuses on further developing understanding and skills in computational thinking such as precisely and accurately describing problems and the use of modular approaches to solutions. It also focuses on engaging students with specialised learning in preparation for vocational training or learning in the senior secondary years.

Students have opportunities to analyse problems and design, implement and evaluate a range of solutions.

Students consider how human interaction with networked systems introduces complexities surrounding access to data of various types.

Students explore data collection methods and use structured data to analyse, visualise, model and evaluate objects and events.

Students learn how to develop multilevel abstractions; identify standard elements, such as searching and sorting in algorithms; and explore the trade-offs between the simplicity of a model and the faithfulness of its representation.

When defining problems students consider the functional and non-functional requirements of a solution through interacting with the 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 are used and controlled, and suggest how policies and practices can be improved to ensure the sustainability and safety of information systems.

When creating solutions individually, collaboratively and interactively for sharing in online environments, students respect the ownership of information.

## Year 9 Learning Area: Technologies – Digital Technologies

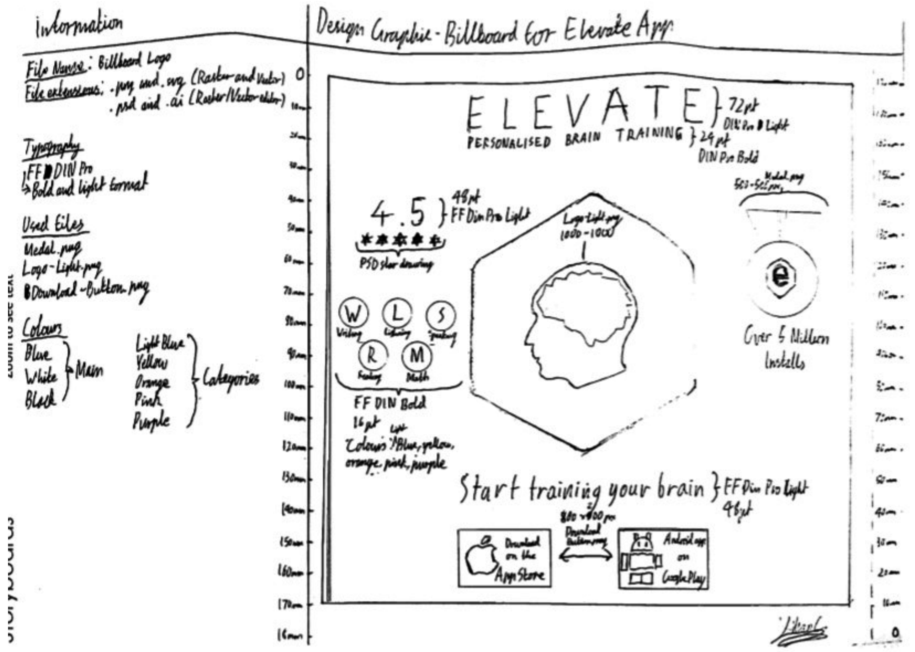
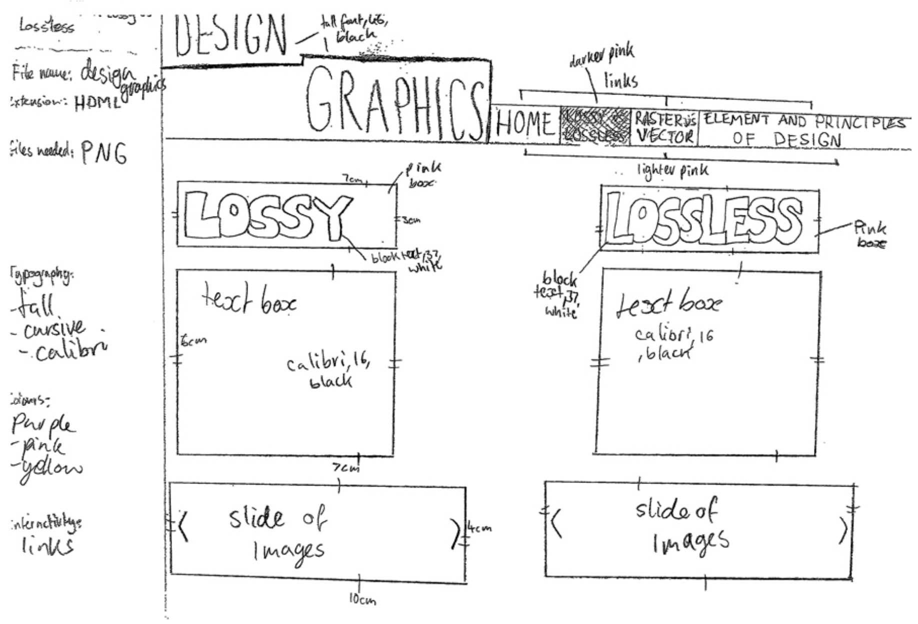
**Year 9 Achievement Standard**

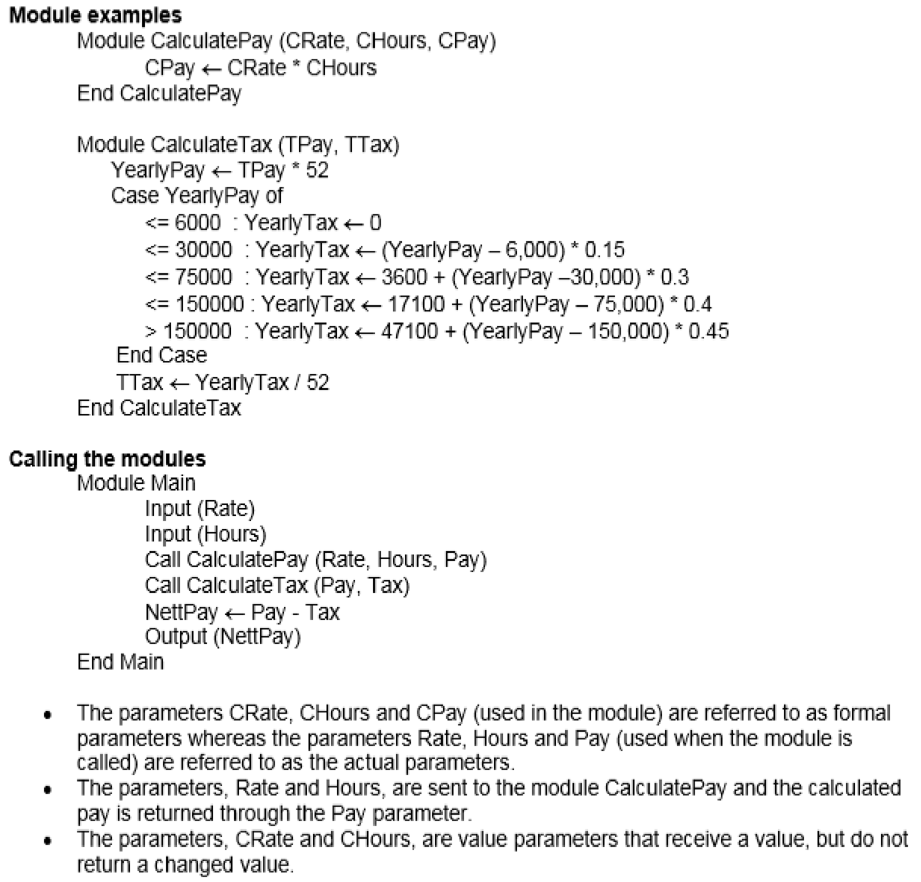
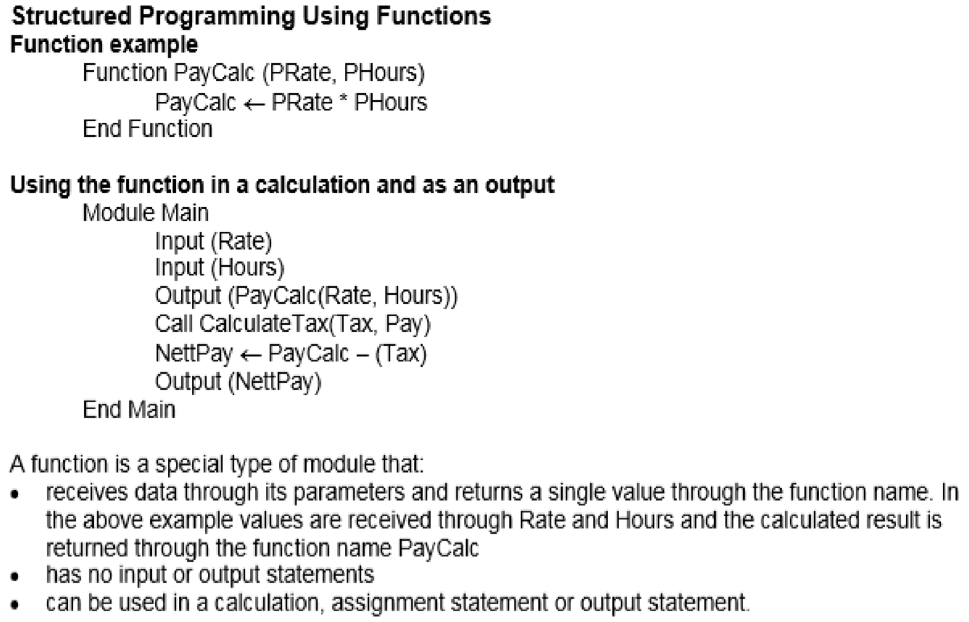
At Standard, students identify the role of hardware and software have in managing, controlling and securing the movement of data in digital systems. They identify different methods used for manipulation, storage and transmission of data. Students explore techniques for acquiring, storing and validating quantitative and qualitative data. They analyse and visualise data to create information and address complex problems. Students create a design for the user experience of a digital system supported by drafts with annotations. They design algorithms, represented diagrammatically and in structured English, and validate plans and programs through tracing. Students implement and apply data storage and organisation techniques. They create and use interactive solutions for sharing ideas and information online, taking into account social contexts.

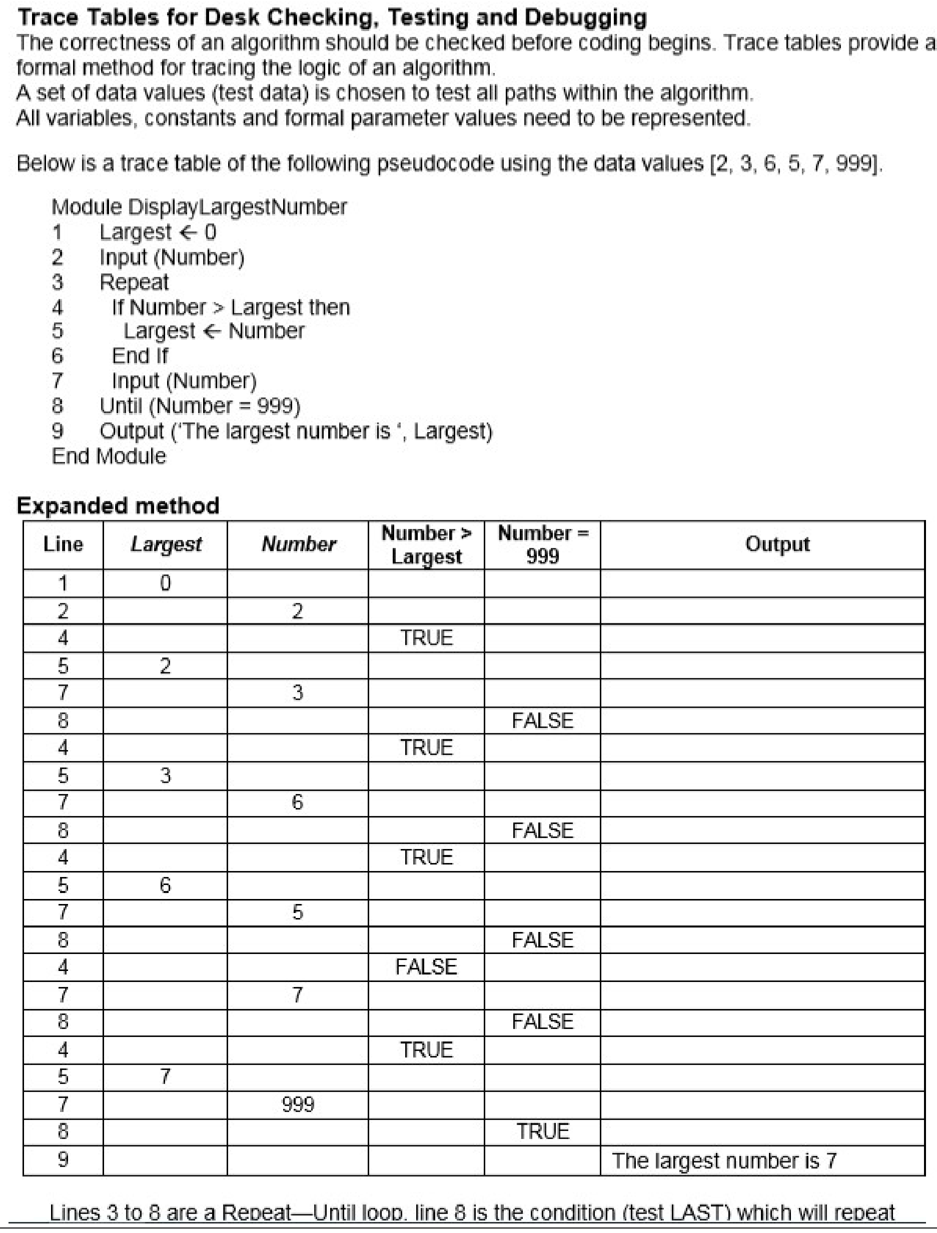
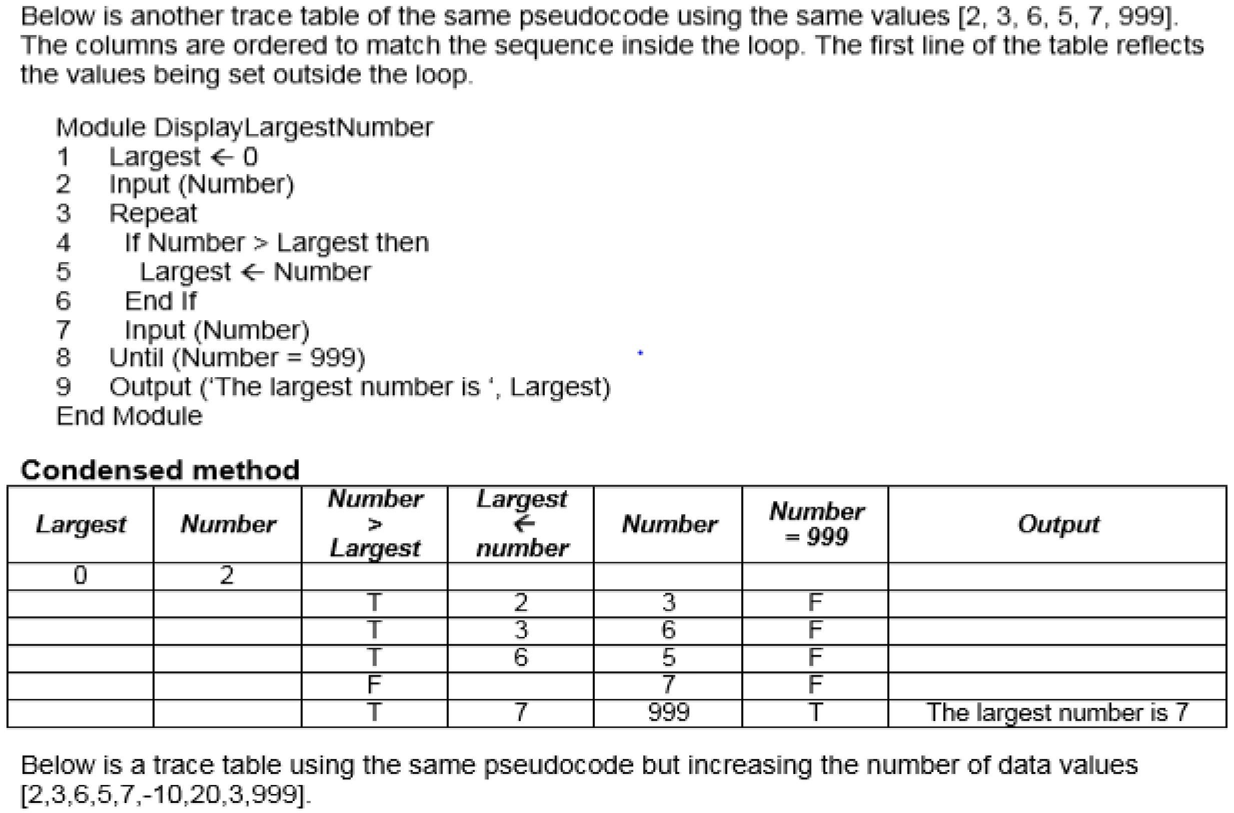
In digital technologies, students identify and define the needs of a stakeholder to create a brief for a solution. They investigate a selection of components/resources to develop ideas, identifying and considering constraints. Students apply design thinking, creativity and enterprise skills. They provide design solutions assessing alternative designs against given criteria, using appropriate technical terms and technology. Students select, test and safely implement appropriate technologies and processes to make solutions. They evaluate design processes against student-developed criteria. Students work independently and collaboratively to manage projects, using digital technology and an iterative and collaborative approach. They consider time, cost, risk and safety.

| Weeks | Syllabus content | Content unpacked | Suggested teaching and learning experiences |
|---|---|---|---|
| 1–3 | **Digital systems**<br>Role of hardware and software in managing, controlling and securing the movement of data in a digital system<br>**Digital implementation**<br>Design the user experience of a digital system<br>Create and use interactive solutions for sharing ideas and information online, taking into account social contexts<br>**Collecting, managing and analysing data**<br>Explore techniques for acquiring, storing and validating quantitative and qualitative data | • Securing and controlling data can be carried out through a variety of physical and virtual tools. These can include the following:<br>▪ Encryption – the process of encoding and decoding a message (data packet), to make it difficult to read if intercepted in transmission<br>▪ Firewalls – a barrier to block unwanted traffic into a system through the identification of authorised IP addresses. Each device connected to a network requires an Internet Protocol (IP) address. Refer to ABC Education What are IP addresses? (https://education.abc.net.au/home#!/media/2198589/what-are-ip-addresses-)<br>▪ MAC filtering – only allows data to be sent and received from network interface cards on devices in a network through the devices' physical addresses<br>▪ Password protection – passwords are used to access a network and to authenticate the user<br>▪ Physical security – Security that is not virtual and tangible in nature. They include biometrics, locks, security cameras, etc.<br>• Common security threats include:<br>▪ DDOS (Distributed Denial of Service) attacks – overwhelming a server by flooding it with internet traffic in order to disrupt its usual operations<br>▪ Intrusion – unauthorised access to a system or network<br>▪ Backdoor attacks – malware used to gain unauthorised access to a system through finding unsecured sections of a network.<br>• *Privacy Act 1988* (Privacy Act): the law that governs the collection, use and disclosure of personal information.<br>• Quantitative: collection of facts and figures, e.g. 12/15.<br>• Qualitative: information that is topic-based discovery, e.g. colour of cars in the car park.<br>• Data sources include: observation, questionnaire, interview, sample forms. | Teacher gathers articles on network security breaches. Students choose a breach and analyse the course and consequences of the breach.<br>Teacher delivers explicit instruction lessons on:<br>1. Network Security tools<br>2. Network Security threats<br>Web design task: Security threats and Solutions for both PC and Network (gathering statistics). Students are required to use their skills of HTML and CSS to create a website that shows students all about networking threats and solutions. Students are marked on their ability to write HTML and CSS, as well as design skills and the content presented.<br>Suggested pages:<br>• Introduction<br>• Security threats (with statistics)<br>• Security solutions (with statistics)<br>• References.<br>Students must write in HTML and CSS. Students must comment on all lines of code in order to receive marks. Images can only be used if referenced and licensed for reproduction. Students must create a design brief before they begin coding.<br>Students revise symbols for network diagrams. Students make sample network diagrams of their current house set-up and their school.<br>Design a network to a client's specifications and analyse connectivity effectiveness. Explain devices etc. Create Network diagram solutions for a small to medium business. Identify the components, devices and the security of the network.<br>In pairs, students visit https://www.oaic.gov.au/privacy/australian-privacy-principles/. Each pair summaries one of the thirteen privacy principles to explain to the class. Focus on "how does it effect me?".<br>'Why is the Privacy Act important to business?' – In teams, students create a short animation to explain to a small Australian business why the Privacy Act is important and the consequences of breaching this Act. This should include gathering data about local business and their knowledge of the Privacy Act.<br>Useful links:<br>• ABC Education How is your personal information kept secret on the internet? (https://education.abc.net.au/home#!/media/2211736/how-is-your-personal-information-kept-secret-on-the-internet)<br>• Digital Technologies Hub Network Security (https://www.digitaltechnologieshub.edu.au/resourcedetail?id=24b84698-09f9-6792-a599-ff0000f327dd#/) |

| Weeks | Syllabus content | Content unpacked | Suggested teaching and learning experiences |
|---|---|---|---|
| | | | • [Khan Academy Journey into Cryptography (https://www.khanacademy.org/computing/computer-science/cryptography)](https://www.khanacademy.org/computing/computer-science/cryptography)<br>• [Digital Technologies Hub Connected via a network (https://www.digitaltechnologieshub.edu.au/teachers/scope-and-sequence/9-10/digital-systems/connected-via-a-network)](https://www.digitaltechnologieshub.edu.au/teachers/scope-and-sequence/9-10/digital-systems/connected-via-a-network) |
| 4–5 | **Investigating and defining**<br>Identify and define the needs of a stakeholder, to create a brief, for a solution<br>Investigate a selection of components/resources to develop solution ideas, identifying and considering constraints<br>**Collaborating and managing**<br>Work independently, and collaboratively to manage projects, using digital technology and an iterative and collaborative approach. Considers time, cost, risk and safety<br>**Designing**<br>Apply design thinking, creativity and enterprise skills | • Project briefs can be created to identify the following aspects of a project:<br>▪ scope<br>▪ intended purpose/client's needs<br>▪ hardware required<br>▪ software required<br>▪ time plan – Gantt chart<br>▪ constraints<br>▪ cost<br>▪ risk<br>▪ legal responsibilities<br>▪ etc.<br>• Components and resources should be identified and their constraints noted. These include both hardware and software and allow for justification where possible.<br>• Iterative in development is identified by a focus on the development of a system with constant end user or client feedback that does not follow a sequential traditional development ideology. Flexibility is gained in this approach and, for experienced developers, an iterative approach can save time and money. | Students create basic timelines in Gantt chart format. These can be based on schoolwork to complete by the end of the term/semester or a one-month holiday in Europe. With travel times, location and what holiday activities can be completed concurrently on the same day.<br>Students research two of their favourite PC games. They then justify the difference between recommended hardware/software and minimum requirements. Students explain why the developers give details of both.<br>**Programming – Part A Design Brief App creator**<br>This design brief is the start of your app development. Your app needs to be based on the following requirements:<br>• environmental sustainability<br>• recycling<br>• e-waste<br>• protection of animals<br>• promoting Aboriginal culture.<br>The design brief will consist of the following:<br>• **brainstorm (bubbl.us in small groups)**<br>• **scope (purpose and why you choose the idea you did)**<br>• **timeline (Gantt chart)**<br>• **resources needed (hardware and software)**<br>• **annotated storyboards (every screen).** |
| 6 | **Designing**<br>Apply design thinking, creativity and enterprise skills<br>Design solutions assessing alternative designs against given criteria, using appropriate technical terms and technology<br>**Digital implementation**<br>Design the user experience of a digital system | • Storyboards – graphical representation of the visual elements of a user interface. Used also in the animation and film design process.<br>• Storyboards should include interactivity, functionality, navigation, useability elements, colours, sizes, etc.<br>• Functionality is the inclusion of a range of different operations and how well they perform to attract and retain users.<br>• Accessibility is making an interface as reachable to as many users as possible, including different skills, abilities, platforms and internet capabilities.<br>• Useability is the ease in which an interface can be accessed or its effectiveness in achieving its intended purpose. | • Select a user interface where students can create a storyboard based on the current interface. This can be a popular website, the school's website or an app of choice. Then students redesign the interface to make it more functional, accessible and with increased useability.<br>Useful link:<br>• The World's Worst Website Ever! (https://www.theworldsworstwebsiteever.com/)<br>**Programming – Part A: Design Brief App creator**<br>This design brief is the start of your app development. Your app needs to be based on the following requirements:<br>• environmental sustainability<br>• recycling<br>• e-waste<br>• protection of animals<br>• promoting Aboriginal culture.<br>The design brief will consist of the following:<br>• **brainstorm (bubbl.us in small groups)**<br>• **scope (purpose and why you choose the idea you did)**<br>• **timeline (Gantt chart)**<br>• **resources needed (hardware and software)**<br>• **annotated storyboards (every screen).** |

| Weeks | Syllabus content | Content unpacked | Suggested teaching and learning experiences |
|---|---|---|---|
| | | <br> Information<br>File Name: Billboard Logo<br>File extensions: .png and .svg (Raster and Vector)<br>.psd and .ai (Raster/Vector editor)<br>Typography<br>FF DIN Pro<br>Bold and light format<br>Used Files<br>Medal.png<br>Logo-Light.png<br>Download-Button.png<br>Colours<br>Blue<br>White<br>Black<br>Main<br>Light Blue<br>Yellow<br>Orange<br>Pink<br>Purple<br>Categories<br>Design Graphic - Billboard for Elevate App<br>ELEVATE<br>72pt<br>DIN Pro Light<br>PERSONALISED BRAIN TRAINING<br>24pt<br>DIN Pro Bold<br>4.5<br>48pt<br>FF Din Pro Light<br>PSD star drawing<br>Logo-Light.png<br>1000-1000<br>Medal.png<br>500-500px<br>W<br>L<br>S<br>R<br>M<br>FF DIN Bold<br>16pt<br>Colours: Light Blue, yellow, orange, pink, purple<br>Over 5 Million Installs<br>Start training your brain<br>FF Din Pro Light<br>48pt<br>Download Button.png<br>Download on the App Store<br>Android app on Google Play<br><br> Lossless<br>File name: design graphics<br>Extension: HDML<br>Files needed: PNG<br>Typography:<br>-tall<br>- cursive<br>- calibri<br>Colours:<br>Purple<br>-pink<br>-yellow<br>interactivity links<br>DESIGN<br>tall font, 66, black<br>GRAPHICS<br>darker pink<br>links<br>HOME<br>LOSSY VS LOSSLESS<br>RASTER VS VECTOR<br>ELEMENT AND PRINCIPLES OF DESIGN<br>lighter pink<br>7cm<br>pink box<br>LOSSY<br>3cm<br>block text, 37, white<br>LOSSLESS<br>Pink box<br>block text, 37, white<br>text box<br>6cm<br>calibri, 16, black<br>text box<br>calibri, 16, black<br>7cm<br>slide of Images<br>4cm<br>10cm<br>slide of Images | |

| Weeks | Syllabus content | Content unpacked | Suggested teaching and learning experiences |
|---|---|---|---|
| 7–9 | **Digital implementation**<br>Design algorithms, represented diagrammatically and in structured English, and validate plans and programs through tracing | <br>**Module examples**<br>        Module CalculatePay (CRate, CHours, CPay)<br>                CPay ← CRate * CHours<br>        End CalculatePay<br><br>        Module CalculateTax (TPay, TTax)<br>            YearlyPay ← TPay * 52<br>            Case YearlyPay of<br>                <= 6000  : YearlyTax ← 0<br>                <= 30000  : YearlyTax ← (YearlyPay – 6,000) * 0.15<br>                <= 75000  : YearlyTax ← 3600 + (YearlyPay –30,000) * 0.3<br>                <= 150000 : YearlyTax ← 17100 + (YearlyPay – 75,000) * 0.4<br>                > 150000  : YearlyTax ← 47100 + (YearlyPay – 150,000) * 0.45<br>            End Case<br>            TTax ← YearlyTax / 52<br>        End CalculateTax<br><br>**Calling the modules**<br>        Module Main<br>                Input (Rate)<br>                Input (Hours)<br>                Call CalculatePay (Rate, Hours, Pay)<br>                Call CalculateTax (Pay, Tax)<br>                NettPay ← Pay - Tax<br>                Output (NettPay)<br>        End Main<br><br>• The parameters CRate, CHours and CPay (used in the module) are referred to as formal parameters whereas the parameters Rate, Hours and Pay (used when the module is called) are referred to as the actual parameters.<br>• The parameters, Rate and Hours, are sent to the module CalculatePay and the calculated pay is returned through the Pay parameter.<br>• The parameters, CRate and CHours, are value parameters that receive a value, but do not return a changed value.<br><br><br>**Structured Programming Using Functions**<br>**Function example**<br>        Function PayCalc (PRate, PHours)<br>                PayCalc ← PRate * PHours<br>        End Function<br><br>**Using the function in a calculation and as an output**<br>        Module Main<br>                Input (Rate)<br>                Input (Hours)<br>                Output (PayCalc(Rate, Hours))<br>                Call CalculateTax(Tax, Pay)<br>                NettPay ← PayCalc – (Tax)<br>                Output (NettPay)<br>        End Main<br><br>A function is a special type of module that:<br>• receives data through its parameters and returns a single value through the function name. In the above example values are received through Rate and Hours and the calculated result is returned through the function name PayCalc<br>• has no input or output statements<br>• can be used in a calculation, assignment statement or output statement. | • Students are to create algorithms and/or flow charts for a variety of common programs.<br>    ▪ Basic tax calculator<br>    ▪ Cricket score calculator<br>    ▪ Choosing clothes for a character<br>    ▪ PacMan game<br>    ▪ etc.<br>• Students then use a Trace Table to validate their programs and analyse the outputs.<br><br>**Programming – Part B: Flowcharts and pseudocode**<br>This design brief is the start of your app development. Your app needs to be based on the following requirements:<br>• environmental sustainability<br>• recycling<br>• e-waste<br>• protection of animals<br>• promoting Aboriginal culture.<br><br>After the completion of the design brief, students are required to produce a flowchart and to pseudocode (diagram and algorithms) their intended solution.<br><br>Year 9 Sample assessment task, That's Amazing (https://k10outline.scsa.wa.edu.au/home/assessment/assessment-activities/year9) |

| Weeks | Syllabus content | Content unpacked | Suggested teaching and learning experiences |
|---|---|---|---|
| 10–11 | **Digital implementation**<br>Design algorithms, represented diagrammatically and in structured English, and validate plans and programs through tracing | • Validation of algorithms is generally completed through desk checking or tracing (Trace Table) the intended purposes based on predetermined inputs. Note: alternatives ways of validating can be considered.<br> | <br>Students should output data to a Trace Table. The template should be provided to students as well as sample algorithms and data to be outputted. The complexity of the algorithms should increase over the examples.<br>**Programming – Part C: Desk checking and validation**<br>After creating algorithms and flowcharts for their intended product, students use a predetermined set of data supplied by the teacher to complete a Trace Table. Students should also list the validation techniques that they will incorporate into their programs. |

**Trace Tables for Desk Checking, Testing and Debugging**

The correctness of an algorithm should be checked before coding begins. Trace tables provide a formal method for tracing the logic of an algorithm.
A set of data values (test data) is chosen to test all paths within the algorithm.
All variables, constants and formal parameter values need to be represented.

Below is a trace table of the following pseudocode using the data values [2, 3, 6, 5, 7, 999].

```
Module DisplayLargestNumber
1    Largest ← 0
2    Input (Number)
3    Repeat
4      If Number > Largest then
5        Largest ← Number
6      End If
7      Input (Number)
8    Until (Number = 999)
9    Output ('The largest number is ', Largest)
End Module
```

**Expanded method**

| Line | *Largest* | *Number* | Number > Largest | Number = 999 | Output |
|---|---|---|---|---|---|
| 1 | 0 | | | | |
| 2 | | 2 | | | |
| 4 | | | TRUE | | |
| 5 | 2 | | | | |
| 7 | | 3 | | | |
| 8 | | | | FALSE | |
| 4 | | | TRUE | | |
| 5 | 3 | | | | |
| 7 | | 6 | | | |
| 8 | | | | FALSE | |
| 4 | | | TRUE | | |
| 5 | 6 | | | | |
| 7 | | 5 | | | |
| 8 | | | | FALSE | |
| 4 | | | FALSE | | |
| 7 | | 7 | | | |
| 8 | | | | FALSE | |
| 4 | | | TRUE | | |
| 5 | 7 | | | | |
| 7 | | 999 | | | |
| 8 | | | | TRUE | |
| 9 | | | | | The largest number is 7 |

Lines 3 to 8 are a Repeat—Until loop. line 8 is the condition (test LAST) which will repeat

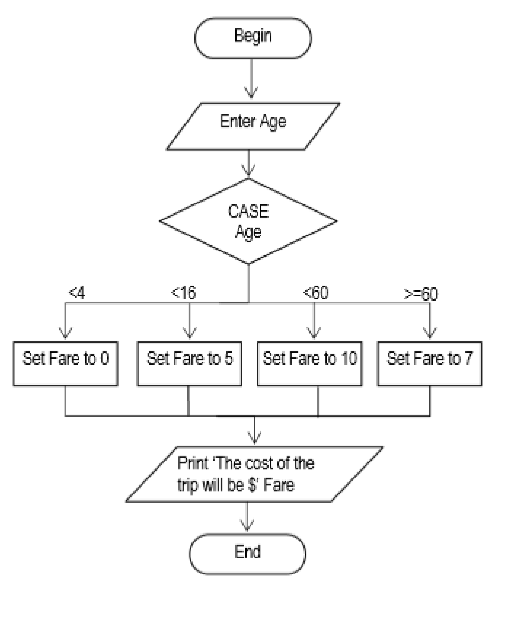
Below is another trace table of the same pseudocode using the same values [2, 3, 6, 5, 7, 999]. The columns are ordered to match the sequence inside the loop. The first line of the table reflects the values being set outside the loop.

```
Module DisplayLargestNumber
1    Largest ← 0
2    Input (Number)
3    Repeat
4      If Number > Largest then
5        Largest ← Number
6      End If
7      Input (Number)
8    Until (Number = 999)
9    Output ('The largest number is ', Largest)
End Module
```

**Condensed method**

| *Largest* | *Number* | *Number > Largest* | *Largest ← number* | *Number* | *Number = 999* | *Output* |
|---|---|---|---|---|---|---|
| 0 | 2 | | | | | |
| | | T | 2 | 3 | F | |
| | | T | 3 | 6 | F | |
| | | T | 6 | 5 | F | |
| | | F | | 7 | F | |
| | | T | 7 | 999 | T | The largest number is 7 |

Below is a trace table using the same pseudocode but increasing the number of data values [2,3,6,5,7,-10,20,3,999].

| Weeks | Syllabus content | Content unpacked | Suggested teaching and learning experiences |
|---|---|---|---|
| | | **Diagrammatical designs can include storyboards and flowcharts.**<br>**Flowchart**<br><br> Begin<br>Enter Age<br>CASE Age<br><4<br><16<br><60<br>>=60<br>Set Fare to 0<br>Set Fare to 5<br>Set Fare to 10<br>Set Fare to 7<br>Print 'The cost of the trip will be $' Fare<br>End<br>**Pseudocode**<br>**Multi way selection (Case)**<br>Input (Age)<br>Case Age of<br>    < 4 : Fare ← 0<br>    < 16 : Fare ← 5<br>    < 60 : Fare ← 10<br>    >= 60 : Fare ← 7<br>End Case<br>Output ('The cost of the trip will be $', Fare) | |
| 12–14 | **Digital implementation**<br>Implement and apply data storage and organisation techniques<br>Create interactive solutions for sharing ideas and information online, taking into account social contexts and legal responsibilities | • At this level, students should be creating solutions to everyday programs in an object-oriented environment. Object-orientated programming (OOP) languages focus on attaching data to objects rather than logical functions. These languages include python, JavaScript, Java, Visual basic, C++, Swift, Ruby, etc.<br>• Sequence of steps refers to the order of a program. This must be logical to avoid any logic errors.<br>• Control Structures are expressions used in programming to make decisions and iterate code.<br>• Branching is a term that refers to having more than one intended outcome.<br>• The digital solution must have choices. These are called selection. The three types of selection are:<br>  ▪ one-way selection (if-then)<br>  ▪ two-way selection (if-then-else)<br>  ▪ multiway selection (CASE).<br>• Iteration:<br>  ▪ test first (do while)<br>  ▪ test last (repeat loop)<br>  ▪ fixed (for loop)<br>• Modularity/Modularisation is the technique of writing smaller independent functions, or sections, that interact and pass parameters between the modules when required. Modularisation allows for the interchangeability of these modules when required. Modularisation is used over programming one long continuous piece of code, as it is time efficient in reusing of code, easier to debug and isolate errors, easier to read and lends itself to collaborative teamwork.<br>• Pseudocode: Software Modularisation (https://youtu.be/DWIgTlaY1SE) | Students are given a list of control structure definitions, including terms such as variables, constants, modularisation, stings, arrays etc. to match to the definitions and separate examples.<br>In the chosen object-orientated programming languages student are required to learn the skills necessary for basic development. These should include all control structures.<br>Year 9 Sample assessment task, That's Amazing (https://k10outline.scsa.wa.edu.au/home/assessment/assessment-activities/year9)<br>**Programming – Part D: Creation of App**<br>This design brief is the start of your app development. Your app needs to be based on the following requirements:<br>• environmental sustainability<br>• recycling<br>• e-waste<br>• protection of animals<br>• promoting Aboriginal culture.<br>Students are to create an App based on the student's design brief.<br>**App software can include, but is not limited to:**<br>• MIT app inventor™<br>• Appy Pie™<br>• Shoutem™<br>• Swiftic™<br>• GoodBarber™<br>• BuildFire™<br>• Mobincube™<br>• AppInstitute™<br>• AppMachine™ |

| Weeks | Syllabus content | Content unpacked | Suggested teaching and learning experiences |
|---|---|---|---|
| 15–16 | **Evaluating**<br>Evaluate design processes and solutions against student developed criteria<br>**Digital implementation**<br>Create and use interactive solutions for sharing ideas and information online, taking into account social contexts and legal responsibilities<br>**Collaborating and managing**<br>Work independently, and collaboratively to manage projects, using digital technology and an iterative and collaborative approach. Considers time, cost, risk and safety | • Evaluation should be conducted throughout the software development process to allow for reflection and future best practice.<br>• Justification against the design process and compared to its intended purpose is critical, not just the finished product.<br>• Collaboration and sharing ideas online can take many forms including Google hangouts, MS teams, Slack, WebEx, Asana, Codingteam, etc.<br>• Online legal responsibilities refer to the following laws:<br>▪ online defamation (https://www.efa.org.au/Issues/Censor/defamation.html)<br>▪ *Privacy Act 1988* (https://www.oaic.gov.au/privacy/the-privacy-act/)<br>▪ *Spam Act 2003* (https://www.acma.gov.au/avoid-sending-spam)<br>▪ piracy and copyright (https://onyxonlinelaw.com/internet-piracy-laws-australia/) | As a class, students evaluate the classes' top five apps currently on their phones.<br>Students are to create a short test with answers on the following terms:<br>• online defamation<br>• *Privacy Act 1988*<br>• *Spam Act 2003*<br>• Piracy and copyright<br>Use Blooms taxonomy and Year 9 Assessment pointers in order to shape the questions.<br>This task can be created in an electronic environment such as Quizlet.com.<br>**Part E: Evaluation of your APP**<br>This task is designed to give you and your peers some quality feedback. We will use this data gathered in order to make decisions on future apps that we create. This should include a self-evaluation for student reflection. This evaluation should be conducted in an online collaborative environment and include a peer and self-assessment.<br>Example of online collaborative tools that can be used for evaluation include:<br>• Bubbl.us<br>• Twiddla<br>• Wikispaces<br>• Wiggio.<br>More information and resources:<br>• https://www.esafety.gov.au/educators/classroom-resources/yes-project<br>• https://foundation.mozilla.org/en/initiatives/web-literacy/<br>• https://www.australia.gov.au/information-and-services/public-safety-and-law/online-safety<br>• https://www.cyber.gov.au/advice |
| 17–18 | **Representation of data**<br>Different methods of manipulation, storage and transmission of data | • Compression is referred to as the reducing of a file in size by changing its properties. Compression aims to reduce file size without significantly reducing the quality of the file. Smaller file sizes have faster transfer speeds. https://searchstorage.techtarget.com/definition/compression<br>• The two main types of compression are lossy and lossless. https://youtu.be/2Qo5prktYNQ<br>▪ Lossy compression – a file is compressed with major loss of data. This data cannot be fully restored. Creates smaller file sizes and quality of image. Formats include JPEG, MPEG, WMA.<br>▪ Lossless compression – there is elimination of small elements of data to compress a file allowing it to be restored to a higher level than a lossy file. Less data is lost, resulting in larger file sizes and higher quality of image. Formats include PNG, Raw, MP4.<br>• Students can use algorithms to identify and reproduce patterns.<br>• Characteristics of simple data types are:<br>▪ integer<br>▪ real (floating point number)<br>▪ boolean<br>▪ character.<br>• Characteristics of complex data types are:<br>▪ string<br>▪ one-dimensional. | • Open a supplied source video in a video transcoder, such as https://handbrake.fr/. Students then save the supplied source video in a variety of different formats to compare and contrast their features, including file size and quality. This is easily represented in a table format. Students can then create a 'how to guide', in the creation of lossy and lossless images.<br>**Digital instruction manual**<br>• Students create a digital instruction manual on a subject of their choice that includes both audio and visual components**.**<br>The task can be conducted in two parts:<br>**Part A:** Students are required to:<br>• plan, design and create the digital instruction manual<br>• use software and hardware together to create the manual; a minimum of five production steps are required.<br>**Part B:** Students are required to:<br>• make at least one choice while interacting with the manual. This could include being guided to a different section of the manual for further topic specific information<br>• evaluate their work against a student-developed criteria.<br>• Students are provided a digital task booklet outlining expectations, including timelines**.**<br>**Note: student examples of work are available on the extranet.**<br>Other resources:<br>• https://www.digitaltechnologieshub.edu.au/teachers/lesson-ideas/seeing-the-big-picture<br>• https://csfieldguide.org.nz/en/chapters/coding-compression/<br>• https://csfieldguide.org.nz/en/chapters/data-representation/ |

| Weeks | Syllabus content | Content unpacked | Suggested teaching and learning experiences |
|---|---|---|---|
| | | | • https://www.digitaltechnologieshub.edu.au/teachers/lesson-ideas/everything-you-always-wanted-to-know<br>• https://australiancurriculum.edu.au/resources/work-samples/samples/digital-project-representation-of-data-below/<br>• https://www.digitaltechnologieshub.edu.au/school-leaders/making-a-difference/what-would-my-preferred-ai-future-look-like |
| 19–20 | **Collecting, managing and analysing data**<br>Analyse and visualise data to create information and address complex problems | • To gather quality data, teachers need to increase students' questioning techniques with examples and practise answers. If data is gathered from third party sources, students and teachers need to analyse the validity and accuracy of the data set.<br>• Analysing data should include finding trends to aid in conclusions, future planning or decision making. With this information, students can predict and justify a future trend such as population growth or birth rates in Australia.<br>• Student should develop and design database solutions. These databases should include:<br>▪ database design, creation, modification, testing and evaluation of a database solution<br>▪ creation of tables, queries, forms and reports<br>▪ use and development of a user interface.<br>• The database management systems should provide the student with opportunity to:<br>▪ create a working relational database<br>▪ construct simple queries using SQL within one or two tables<br>▪ construct queries across multiple tables using a database tool<br>▪ apply programmed control structures<br>▪ develop and use a user interface.<br>• Some suggested database management system software are:<br>▪ Microsoft Access<br>▪ MySQL<br>▪ FileMaker<br>▪ FoxPro<br>▪ Paradox. | **Incorporate data through programming**<br>• Students can use their selected programming language to incorporate data and analyse it.<br>• An example of importing and analysing data with Python can be seen with the link below. https://www.youtube.com/watch?v=Tq6rCWPdXoQ<br>**Making data interesting**<br>• Students use a program such as Cavna.com, to create an infographic. The data set can be large and provided by the teacher. Students compete to see who can make the most appealing infographic with the data given.<br>**Membership database**<br>• Students are required to produce a database for a sporting club or interest group. As a class, students explore concepts related to the collection of data and techniques used for data storage, including privacy and security considerations.<br>• Students also develop knowledge and skills associated with the development of a database using appropriate software.<br>This task can be conducted in two parts:<br>**Part A:** Students are required to:<br>• conduct a survey to find out what member information is commonly collected by sporting or interest groups and the types of hardware and software required to gather and implement this information and data<br>• choose a sporting or interest group to design a membership database<br>• design and create a database allowing for the storage and retrieval of member details, including an acknowledgement of having read and understood the code of conduct, and any additional information as required by the survey results.<br>**Part B:** Students are required to:<br>• conduct the survey and use the information to create a database<br>• explain how the database would assist people collecting information for other sporting or interest groups<br>• define how the database could be modified if there was a requirement to change the details collected in the database.<br>Useful websites:<br>• https://www.digitaltechnologieshub.edu.au/teachers/lesson-ideas/a-spreadsheet-s-secret-weapon<br>• https://support.office.com/en-us/article/Top-ten-ways-to-clean-your-data-2844b620-677c-47a7-ac3e-c2e157d1db19<br>• https://education.abc.net.au/statistics-game/#/<br>• https://www.khanacademy.org/math/probability/scatterplots-a1/creating-interpreting-scatterplots/v/correlation-and-causality<br>• https://www.digitaltechnologieshub.edu.au/teachers/lesson-ideas/who-wants-to-be-a-millionaire |

Note: the above Teaching and Learning Outline is based on two hours per week for 20 weeks for a total of 40 hours.