Government of **Western Australia**
School Curriculum and Standards Authority

Sample Teaching and Learning Outline

Technologies

Digital Technologies

Year 8

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.

2020/6717v4

This document is an introduction to planning a teaching and learning outline with syllabus content for Year 8 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 assessment should be drawn. This *Year 8 Sample Teaching and Learning Outline* provides teachers with possible learning experiences over 18 weeks (two hours per week) and unpacks the syllabus content to assist teachers in their understanding. Teachers may choose from the **Suggested teaching and learning experiences** that suit the context of the school.

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 8 Syllabus Content – Digital Technologies

Content	Description
Digital systems	Methods of data transmission and security in wired, wireless and mobile networks Specifications of hardware components and their impact on network activities
Representation of data	Binary is used to represent data in digital systems
Collecting, managing and analysing data	Evaluate the authenticity, accuracy and timeliness of acquired data Evaluate and visualise data, using a range of software, to create information, and use structured data to model objects or events
Digital implementation	Design the user experience of a digital system Design plans, using a sequence of steps, and represent them diagrammatically and in English, to solve a problem and to predict output for a given input to identify errors Implement and modify solutions, that include user interfaces within a programming environment, including the need for choice of options and/or repeating options Create and communicate interactive ideas collaboratively online, taking into account social contexts
Investigating and defining	Investigate a given need or opportunity for a specific purpose Evaluate and apply a given brief Consider components/resources to develop solutions, identifying constraints
Designing	Design, develop, evaluate and communicate alternative solutions, using appropriate technical terms and technology Produce a simple plan designed to solve a problem, using a sequence of steps

Content	Description
Producing and implementing	Safely apply appropriate techniques to make solutions using a range of components and equipment
Evaluating	Develop contextual criteria independently to assess design processes and solutions
Collaborating and managing	Work independently, and collaboratively when required, to plan, develop and communicate ideas and information when managing processes

Year Level Description

In Year 8, learning in digital technologies focuses on further developing understanding and skills in computational thinking, such as decomposing problems, and engaging students with a wider range of information systems as they broaden their experiences and involvement in national, regional and global activities.

Students have opportunities to create a range of solutions, such as interactive web applications or simulations of relationships between objects in the real world.

Students investigate the properties of networked systems and their suitability and use for the transmission of data types. They acquire, analyse, visualise and evaluate various types of data, and the complexities of storing and transmitting that data in digital systems. Students use structured data to model objects and events that shape the communities they actively engage with. They further develop their understanding of the vital role that data plays in their lives, and how the data and related systems define and are limited by technical, environmental, economic and social constraints.

Students further develop abstractions, identifying common elements, while decomposing apparently different problems and systems to define requirements; and recognise that abstractions hide irrelevant details for particular purposes. When defining problems, students identify the key elements of the problems and the factors and constraints at play. They design increasingly complex algorithms that allow data to be manipulated automatically, and explore different ways of showing the relationship between data elements to help computation. They progress from designing the user interface, to considering user experience factors, such as user expertise, accessibility and usability requirements.

Students have opportunities to plan and manage individual and team projects. They consider ways of managing the exchange of ideas, tasks and files, and techniques for monitoring progress and feedback. When communicating and collaborating online, students develop an understanding of different social contexts; for example, acknowledging cultural practices and meeting legal obligations.

Year 8 Learning Area: Technologies – Digital Technologies

Year 8 Achievement Standard

At Standard, students identify methods of data transmission and security in wired, wireless and mobile networks and identify specifications of hardware components and outline apparent impacts on network activities. They identify how binary is used to represent data in digital systems. Students evaluate the authenticity, accuracy and timeliness of acquired data and use a range of software to evaluate and visualise data. Students present diagrammatically and in English, their designs and plans for the user experience of a digital system, with sequenced steps. They predict output for a given input to identify errors. Students modify and implement digital solutions, considering the user interface within a programming environment and the need for user choice and/or repeating options. They work collaboratively online to create and communicate interactive ideas with consideration for social contexts.

In digital technologies, students investigate a given need or opportunity for a specific purpose. They evaluate and apply a given brief, using some examples. Students consider and select components/resources to develop solutions, identifying constraints. They use appropriate technical terms and technology to design, develop, evaluate and communicate alternative digital solutions. Students develop sequenced steps to produce a simple, problem-solving plan. They apply safe and appropriate techniques to make solutions, using a range of components and equipment. Students independently develop contextual criteria to assess design processes and solutions. They work independently, and collaboratively, to plan, develop and communicate ideas and information when managing projects.

Weeks	Syllabus content	Content unpacked	Suggested teaching and learning experiences
1–2	**Representation of data** Binary is used to represent data in digital systems	• All data is represented as binary [zeros (0) or, ones (1)]. This is due to all computers using a Central Processing Unit and mathematical algorithms as their core functions. • Eight bits equals one byte. • Data is processed in correct order at all times, sequentially and logically. • Data has two states, On and Off, represented by zeros (0) and ones (1). • These are electronic pulses used to drive microchips and other hardware devices.	• Watch the YouTube™ video Techquickie: Binary Numbers and Base Systems as Fast as Possible (https://youtu.be/LpuPe81bc2w). • Students then comprehensively explain why binary is important in computer systems. • Mathematical activities to convert binary to decimal and decimal to binary. Worksheet with eight columns to help with mathematical conversions. • Access a variety of binary games online for reinforcing concepts; for example, Code The Binary Game (https://studio.code.org/projects/applab/iukLbcDnzqgoxuu810unLw). • CS unplugged activities (https://classic.csunplugged.org/binary-numbers/). • Short theory test on binary • Alternative activity: Crack the code to find the hidden treasure (https://k10outline.scsa.wa.edu.au/home/assessment/assessment-activities/year8).
3–5	**Digital systems** Methods of data transmission and security in wired, wireless and mobile networks Specifications of hardware components and their impact on network activities **Digital implementation** Create and communicate interactive ideas collaboratively online, taking into account social contexts	• Local Area Networks (LAN) – interconnected devices that span small geographical locations. • Wide Area Networks (WAN) – interconnected devices that span large geographical locations. • Wireless data transmission includes radio, cellular, satellite and microwave. • Cloud Computing is a service provided by a third party to allow people or businesses to access storage space on a server without setting up physical equipment onsite. Usually has tiers of payment options that are based on storage space and/or services required. • Network threats include viruses, worms, DDOS attacks, back doors, phishing and intrusion. Visit Webopedia: Online Tech Dictionary for Students, Educators and IT Professionals Webopedia.com (https://www.webopedia.com/) for clarification of definitions. • Network security includes encryption, firewalls, WEP keys etc. ▪ Encryption is a security feature used primarily for the secure transferring of data from one device to another. Data is encrypted or scrambled before being sent with a secure code. If data is intercepted it is difficult to be read or used. The data is decrypted at the receiving end using a decryption key that allows the data to be viewed or processed. ▪ Firewalls can be physical or virtual. Firewalls block any foreign or unrecognised Internet Protocol (IP) addresses to create a barrier for intrusion. ▪ WEP keys (Wired Equivalent Privacy) is a set of rules that govern the secure transfer of data wirelessly. This is done through encryption of end-to-end devices. • Network hardware – servers, switches, routers, modems, Integrated Service Routers (ISR), Network Interface Cards (NIC).	• Create a chart with that compares similarities and differences between radio, cellular, satellite and microwave. • Watch the YouTube video Techquickie™: What is "The Cloud" as Fast As Possible (https://youtu.be/dsKIpLKo8AE) and use a worksheet to answer questions based on the video. • Students work in pairs. One member researchers a network threat whilst the other student researches the network solution. They report back to each other on their findings. • Provide a case study of a security breach where students identify the solution and justify what security measures could have helped to prevent the threat. This can be completed in pairs or in small groups through online collaboration tools. • Use a CISCO packet tracer or predetermined symbols of network devices to draw basic networks (both LAN and WAN). • Students are to use a standard floor plan of a house to draw both a wired and wireless network. ▪ Students label the devices and explain their role in a network. ▪ Students include an example of each device with specifications. ▪ Students then justify why they require that particular device in their network, including any cabling, where appropriate, with specifications.

Weeks	Syllabus content	Content unpacked	Suggested teaching and learning experiences
		▪ A server is a computer with large storage space that sits centrally on a network to manage network operations and connected devices. Visit Techquickie Servers vs Desktop PCs as Fast as Possible (https://youtu.be/ByI1PHMcPJQ). ▪ Switches are used to extend ports allowing for multiple devices to connect to one, such as a server. ▪ Routers are smart switches that direct traffic around a network choosing the most efficient path for packets to follow. ▪ Modems modulate and demodulate analog and digital signals. ▪ Integrated Service Routers (ISRs) are a convergent devices that are found in most home networks. ISRs combine both wired and wireless routing functions, modems, firewalls, basics of switches etc. ▪ Network Interface Cards (NIC's) can be either wired or wireless or both. NICs are used to communicate and connect a device to a network. Without a NIC, a device is unable to connect to the network. • Collaboratively communicating online refers to the creation of blogs, shared Google™ documents, Connect™, virtual meetings etc., where students share ideas and work together in an online space. • Specifications refers to a device's statistics about performance, compatibility, size, power usage, features etc. • Social contexts refers to the immediate social setting surrounding students, including values, beliefs, unspoken rules, physical environment etc.	
6–10	**Investigating and defining** Investigate a given need or opportunity for a specific purpose. Evaluate and apply a given brief. Consider components/resources to develop solutions, identifying constraints. **Designing** Design, develop, evaluate and communicate alternative solutions, using appropriate technical terms and technology Produce a simple plan designed to solve a problem, using a sequence of steps.	• Design briefs contain the scope of the product, the resources needed, the intended target audience and timing of the product. • Flowcharts are designs that allow for a graphical representation of a program with all required elements, including inputs, outputs, process, iteration, selection, variable naming etc. • Programs will require the following control structures: ▪ selection – one-way, two-way, multi-way ▪ iteration: ○ test first (do while) ○ test last (repeat loop) ○ fixed (for loop) ▪ variables ▪ constants ▪ sequence. • Designs in Digital Technologies include storyboards, structure charts, flowcharts, algorithms, pseudocode, network diagrams, wire-frames etc. • The three types of errors in programming are: ▪ logic – the order or sequence is incorrect, such as calling a variable before it is defined. ▪ syntax – the rules of the programming language have not been followed. Use of an incorrect or foreign word or symbol. ▪ runtime – where the program cannot be executed due to being asked something impossible; for example, divide by 0. • User experience in a digital system refers to how the user inputs data and receives processed outputs. These need to be logical in navigation, high in useability and involve validation techniques where applicable.	• Students design and create a simple program that uses branching (selection) and repeating (iteration). This could include adding the total score of a cricket team, students' assessment scores, or recording scores of a video game with a logical sequence of steps. ▪ Students are given a list of programs and choose one to create. These options should have differentiated learning options. ▪ Students must develop the scope, constraints, target audience, resources and timing of their chosen program. • Students will need to design their program using a flowchart, including branching, options and repeating. Students aid the teacher in creating word walls when technical terminology is used or add definitions to their personal glossaries. • Students are given a variety of premade algorithm examples. These can be in written form or as flowcharts. ▪ Students are to use recommended inputs, such as numbers, to test the accuracy of the algorithms. This can be in the form of a trace table. • Some examples must have errors to allow students to identify where the program has an error and/or how to fix the error.

<table>
<tr><th>Weeks</th><th>Syllabus content</th><th>Content unpacked</th><th>Suggested teaching and learning experiences</th></tr>
<tr>
<td></td>
<td>

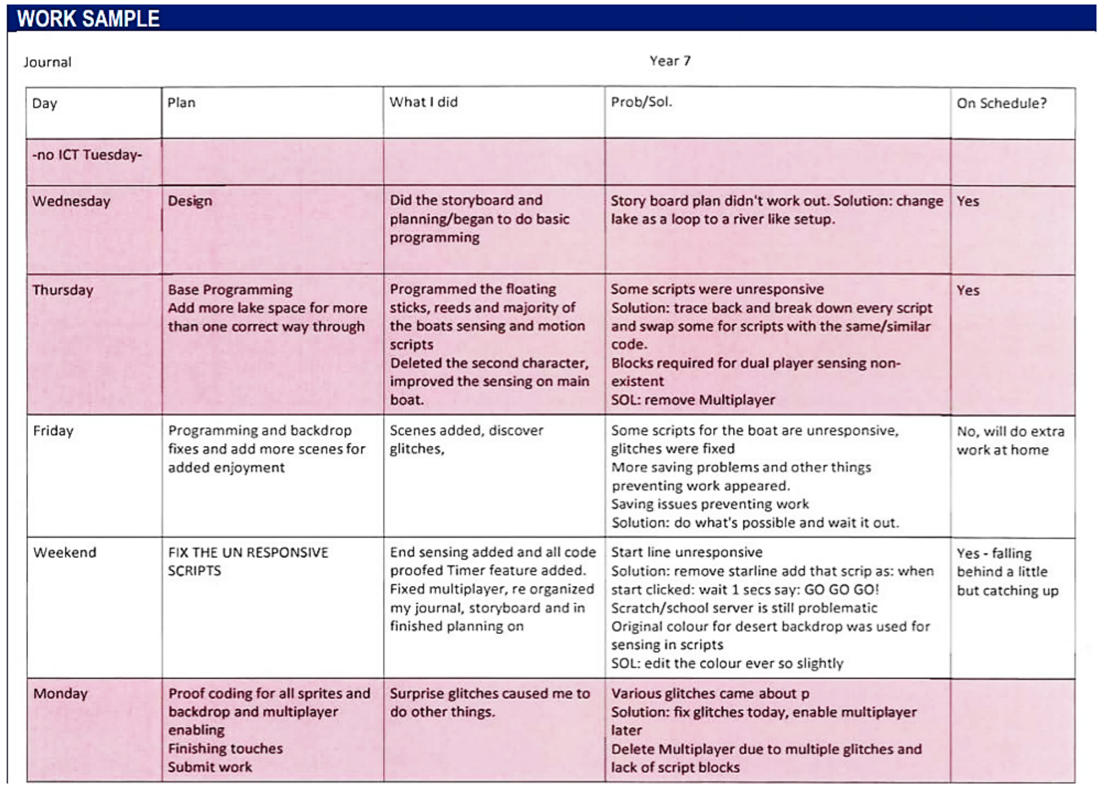
Digital implementation
Design the user experience of a digital system.

Design plans, using a sequence of steps, and represent them diagrammatically and in English, to solve a problem and to predict output for a given input to identify errors.

</td>
<td>

- Validation techniques include limiting a phone number to less than 11 numbers or having two password forms that must match.

Basic student work sample of a time plan (from the School Curriculum and Standards Authority Extranet – Year 7 Judging Standards).

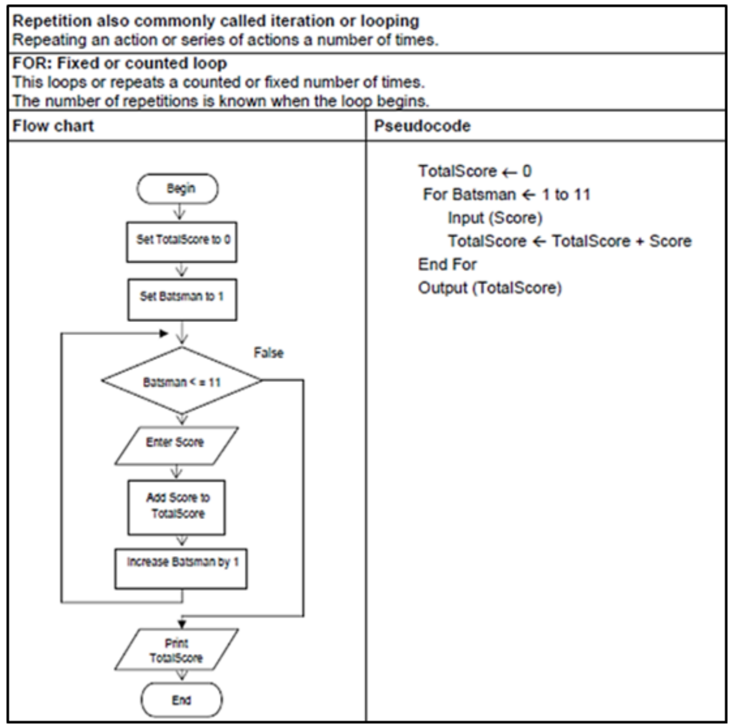


WORK SAMPLE

Journal Year 7

<table>
<tr><th>Day</th><th>Plan</th><th>What I did</th><th>Prob/Sol.</th><th>On Schedule?</th></tr>
<tr><td>-no ICT Tuesday-</td><td></td><td></td><td></td><td></td></tr>
<tr><td>Wednesday</td><td>Design</td><td>Did the storyboard and planning/began to do basic programming</td><td>Story board plan didn't work out. Solution: change lake as a loop to a river like setup.</td><td>Yes</td></tr>
<tr><td>Thursday</td><td>Base Programming
Add more lake space for more than one correct way through</td><td>Programmed the floating sticks, reeds and majority of the boats sensing and motion scripts
Deleted the second character, improved the sensing on main boat.</td><td>Some scripts were unresponsive
Solution: trace back and break down every script and swap some for scripts with the same/similar code.
Blocks required for dual player sensing non-existent
SOL: remove Multiplayer</td><td>Yes</td></tr>
<tr><td>Friday</td><td>Programming and backdrop fixes and add more scenes for added enjoyment</td><td>Scenes added, discover glitches,</td><td>Some scripts for the boat are unresponsive, glitches were fixed
More saving problems and other things preventing work appeared.
Saving issues preventing work
Solution: do what's possible and wait it out.</td><td>No, will do extra work at home</td></tr>
<tr><td>Weekend</td><td>FIX THE UN RESPONSIVE SCRIPTS</td><td>End sensing added and all code proofed Timer feature added. Fixed multiplayer, re organized my journal, storyboard and in finished planning on</td><td>Start line unresponsive
Solution: remove starline add that scrip as: when start clicked: wait 1 secs say: GO GO GO!
Scratch/school server is still problematic
Original colour for desert backdrop was used for sensing in scripts
SOL: edit the colour ever so slightly</td><td>Yes - falling behind a little but catching up</td></tr>
<tr><td>Monday</td><td>Proof coding for all sprites and backdrop and multiplayer enabling
Finishing touches
Submit work</td><td>Surprise glitches caused me to do other things.</td><td>Various glitches came about p
Solution: fix glitches today, enable multiplayer later
Delete Multiplayer due to multiple glitches and lack of script blocks</td><td></td></tr>
</table>

Flowchart example with repetition



Repetition also commonly called iteration or looping
Repeating an action or series of actions a number of times.

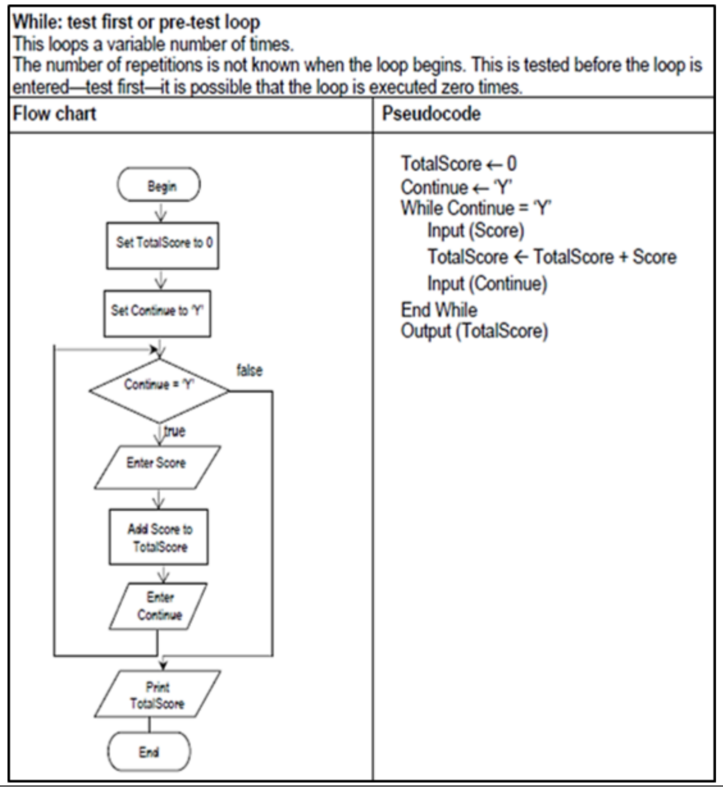
FOR: Fixed or counted loop
This loops or repeats a counted or fixed number of times.
The number of repetitions is known when the loop begins.

Flow chart

Pseudocode

```
TotalScore ← 0
 For Batsman ← 1 to 11
     Input (Score)
     TotalScore ← TotalScore + Score
End For
Output (TotalScore)
```

</td>
<td></td>
</tr>
</table>

Weeks	Syllabus content	Content unpacked	Suggested teaching and learning experiences
		Flowchart example with repetition  While: test first or pre-test loop This loops a variable number of times. The number of repetitions is not known when the loop begins. This is tested before the loop is entered—test first—it is possible that the loop is executed zero times. Flow chart Begin Set TotalScore to 0 Set Continue to 'Y' Continue = 'Y' false true Enter Score Add Score to TotalScore Enter Continue Print TotalScore End Pseudocode TotalScore ← 0 Continue ← 'Y' While Continue = 'Y' Input (Score) TotalScore ← TotalScore + Score Input (Continue) End While Output (TotalScore)	
11–14	**Digital implementation** Implement and modify solutions, that include user interfaces within a programming environment, including the need for choice of options and/or repeating options **Collaborating and managing** Work independently, and collaboratively when required, to plan, develop and communicate ideas and information when managing processes **Producing and implementing**	• Programs will require the following control structures: ▪ selection – one-way, two-way, multi-way ▪ iteration: ○ test first (do while) ○ test last (repeat loop) ○ fixed (for loop) ▪ variables ▪ constants ▪ sequence • A prototype is a small unfinished version of a program used to test a concept or functionality whilst gathering feedback before committing to the final product. • Modification of solutions should be completed continuously throughout the development of the prototype. • Programming skills to be taught: ▪ declare variables ▪ input data in a variety of options ▪ output data ▪ correct syntax ▪ set data types ▪ arithmetic operators ▪ conditions ▪ iteration • formatting basics.	• Students use a programming language that is familiar to both students and teacher, for example, Python, JavaScript, Swift, PHP etc. to produce and prototype a product that suits the requirements of the design. Can be created in pairs or individually. • Students create a simple program based on the design created, that uses branching (selection) and repeating (iteration). This could include adding the total score of a cricket team, students' assessment scores or recording scores of a video game. Record all variations or problems faced throughout the development for evaluation purposes.

Weeks	Syllabus content	Content unpacked	Suggested teaching and learning experiences
	Safely apply appropriate techniques to make solutions using a range of components and equipment		
15–16	**Evaluating** Develop contextual criteria independently to assess design processes and solutions **Collecting, managing and analysing data** Evaluate the authenticity, accuracy and timeliness of acquired data. Evaluate and visualise data, using a range of software, to create information, and use structured data to model objects or events.	• Authenticity refers to data or information that is gathered from a well-respected source. • Accuracy refers to how correct the data or information is with valid research to inform this decision. • Timeliness refers to how current or up-to-date the data or information received is.	• Students evaluate the design and end product of each other's programs. This should be completed via online surveys. Use this as your data set. Students independently develop and apply criteria for evaluation. ▪ Model data sets through individual and class data. ▪ Graph and analyse data from both sources against each other. • The teacher selects a variety of articles from the internet. Students perform web research to confirm and evaluate the articles authenticity, accuracy and timeliness.

Note: the above Teaching and Learning Outline is based on two hours per week for 16 weeks for a total of 32 hours.