Assessment Schedule – 2022

Digital Technologies and Hangarau Matihiko: Analyse an area of computer science (91908)

Assessment Criteria

Achievement	Achievement with Merit	Achievement with Excellence
Analysing *an area of computer science* involves explaining: • the key aspects of the computer science area • relevant algorithms or other mechanisms behind the area • how the area is used, is implemented, or occurs, giving examples • key problems or issues related to the area and how these have been or may be addressed.	***In-depth analysis*** *of an area of computer science* involves: • providing a detailed explanation of how the technical capabilities and limitations of the area relate to humans, giving examples • comparing and contrasting different perspectives on the area.	***Critically analysing*** *an area of computer science* involves: • drawing insightful conclusions about the computer science area.

Cut Scores

Not Achieved	Achievement	Achievement with Merit	Achievement with Excellence
0 – 2	3 – 4	5 – 6	7 – 8

Evidence

N1	N2	A3	A4	M5	M6	E7	E8
Makes relevant comments in some parts of the response, but not enough to holistically show understanding.	Makes relevant comments in some parts of the response, but not enough to holistically show understanding.	Explains: • key aspects of the chosen computer science area • relevant algorithms or mechanisms that support the area • how the area is used or implemented, or occurs, giving examples • key problems or issues related to the area, and how these have been, or may be, addressed.	Explains: • key aspects of the chosen computer science area • relevant algorithms or mechanisms that support the area • how the area is used or implemented, or occurs, giving examples • key problems or issues related to the area, and how these have been, or may be, addressed.	Explains, in detail, how the technical capabilities and limitations of the computer science area relate to humans, giving examples. Compares and contrasts different perspectives on the area.	Explains, in detail, how the technical capabilities and limitations of the computer science area relate to humans, giving examples. Compares and contrasts different perspectives on the area.	Draws insightful conclusions about the computer science area.	Draws insightful conclusions about the computer science area.
Responses are mostly incorrect.	Some incorrect responses.	Some aspects of the response may be partial or weak.		Some aspects of the response may be partial or weak.		Some aspects of the response may be partial or weak.	

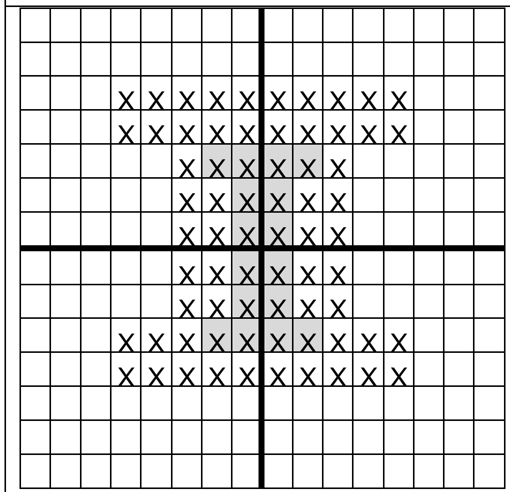
N0 = No response; no relevant evidence.

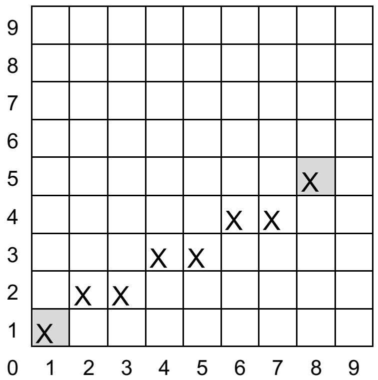
Formal Languages

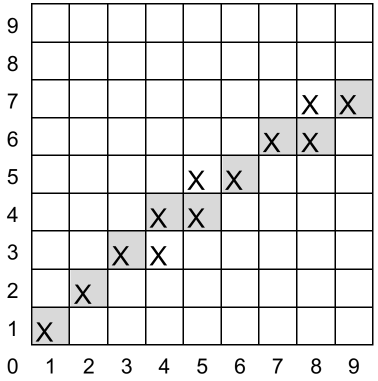
<table>
<tr><th>Question</th><th>Sample evidence</th><th>Achievement</th><th>Achievement with Merit</th><th>Achievement with Excellence</th></tr>
<tr><td>ONE
(a) (i)</td><td>Completes Table
Q = {S1, S2, S3, S4} or S1, S2, S3, S4
Σ = {w, x, y, z}
δ = :
<table>
<tr><td>current state</td><td>input symbol</td><td>new state</td></tr>
<tr><td>S1</td><td>W</td><td>S2</td></tr>
<tr><td>S2</td><td>W</td><td>S2</td></tr>
<tr><td>S2</td><td>X</td><td>S1</td></tr>
<tr><td>S2</td><td>Y</td><td>S4</td></tr>
<tr><td>S3</td><td>W</td><td>S1</td></tr>
<tr><td>S3</td><td>X</td><td>S4</td></tr>
<tr><td>S4</td><td>Z</td><td>S3</td></tr>
</table>
q_0 = S1
F = S4</td><td>Correct answer.</td><td></td><td></td></tr>
<tr><td>(ii)</td><td>(4) WWWWYZ</td><td>Correct answer.</td><td></td><td></td></tr>
<tr><td>(iii)</td><td>String ends with y or x</td><td>Recognises Ends on state 3 of the FSM needs to finish with y or x.</td><td></td><td></td></tr>
<tr><td>(b) (i)</td><td>Can, Ran, Fan</td><td>Correct answer.</td><td></td><td></td></tr>
<tr><td>(ii)</td><td>Answers may vary. An acceptable response might be “must contain a C, F, or R character followed by ‘an’”.</td><td>A number of answers can be accepted, but they must be explained and valid.</td><td></td><td></td></tr>
<tr><td>(iii)</td><td>b[aeiou]bble (lots of different ways)</td><td>Correct answer.</td><td></td><td></td></tr>
<tr><td>(iv)</td><td>Justifies REGEX used, e.g. ‘b[aeiou]bble – must start with a ‘b’, be followed with a vowel, and end in ‘bble’.</td><td>A number of answers can be accepted, but they must be explained and valid.</td><td></td><td></td></tr>
<tr><td>(c) (i)</td><td>E, N</td><td>Correct answer.</td><td></td><td></td></tr>
<tr><td>(ii)</td><td>0, *, +, -, 0-9 or 0, 1, 2, 3, 4, 5, 6, 7, 8, 9</td><td>Correct answer.</td><td></td><td></td></tr>
<tr><td>(iii)</td><td>(7 + 3) * 2</td><td>Correct answer.</td><td></td><td></td></tr>
<tr><td>(iv)</td><td>Answer explains how (N=N) *N was arrived at and then can substitute in the values to solve (7+3) X 2.</td><td>Explains how equation is built up.</td><td></td><td></td></tr>
</table>

Question	Sample evidence	Achievement	Achievement with Merit	Achievement with Excellence
(d)	Regular language can have a finite state machine. Some cases do not allow for a finite state machine e.g. counting, memorising, recognising, and opening and closing brackets (multiple).	Provides response that links regular language to a finite state machine.	Shows understanding of where regular languages are not able to be expressed in a finite state machine, but can resolve them using a context-free language. Explains, in detail, how the technical capabilities and limitations of formal languages relate to humans, giving examples. Compares and contrasts different perspectives on formal languages.	
(e)	Finite number of states is not long enough, as the 0s can be any number long (arbitrary). The first half could be arbitrarily long, so a finite number of states is not enough. In order to check the corresponding 1s you would need the machine to remember the number of 0s. In order to check you would need the machine to 'count'. It cannot do that as it doesn't have memory – FSM have very limited memory.	Provides response that links regular language to a finite state machine.	Recognises at least one example of the evidence and explains it. Explains, in detail, how the technical capabilities and limitations of formal languages relate to humans, giving examples. Compares and contrasts different perspectives on formal languages.	
(f)	It can be used to define simple gameplay behaviour in a rather intuitive way: using states and transition conditions from state to state. Once the different states and transitions are known and tested, dead states or trap states can be identified and resolved. Then, you can begin to program the outcome knowing it will work as intended.		Defines states that can only exist at a single stage. Identifies trap states.	Examples of other finite state machines and how they are or could be used. Any of: • innovative and imaginative connections • exploration of less obvious implications • making justified predictions · suggesting improvements • making justified generalisations that could be applied beyond the area itself • use of higher-level thinking skills such as synthesis.
(g) (i)	(10, an 01,001,0001 0^+1)	Correct answer.		
(ii)	Dead states not shown. There is a trap state.	Can identify a dead state or can explain what happens to the unshown values at each state.	Explains why dead states are not shown.	

Computer Graphics

Question	Sample evidence	Achievement	Achievement with Merit	Achievement with Excellence
TWO (a) (i)	Graphics programs often perform all kinds of calculations on the vertices of an object before finally drawing that object on screen. Translation, scaling, and rotation can all be performed on a single shape just by using ‘translate’, ‘scale’, and ‘rotate’, with the shape’s vertices. However, performing many calculations on many vertices can be time-consuming, which is why graphics programmers often use matrix maths to transform shapes.	Identifies two or more examples of what matrix transformations are used for in computer graphics.		
(ii)	Matrices in graphics programming can represent any number of transformations with a single matrix. For example, a single matrix can contain all the values you need to simultaneously translate, scale, and rotate a shape. To do this, you fill the matrix with the appropriate values and then multiply the matrix by all of the shape's vertices. Of course, the trick is to know what values to place in the matrix.	Identifies why matrix transformations are used and gives an example.		
(b) (i)		Applies correct transformation.		
(ii)	$\begin{bmatrix} 2 & 0 \\ 0 & 2 \end{bmatrix}$ $\begin{bmatrix} 2 \\ 3 \end{bmatrix}$ $\begin{bmatrix} 4 \\ 6 \end{bmatrix}$	Correct calculation. Explains calculation.		

<table>
<tr><th>Question</th><th>Sample evidence</th><th>Achievement</th><th>Achievement with Merit</th><th>Achievement with Excellence</th></tr>
<tr><td>(c) (i)</td><td>Computer screens are made of pixels. In order to draw a slope, the computer must fill pixels in. This works fine for continuous lines but causes staircasing or jagged edges on sloping lines. The computer must perform a calculation and then decide which pixel to fill in in order to make the line appear as close to those coordinates as possible. On a computer, the calculations need to be done for every pixel, and if you use the wrong method, it will take too long. For example, mx+c performs a multiplication and addition for each pixel, and multiplication is slow on computers compared with additions, so the image will be displayed slowly or a live animation will appear jerky.</td><td>Correct explanation. Identifies pixels and does a calculation to determine which pixel to fill.</td><td></td><td></td></tr>
<tr><td>(ii)</td><td>
<table>
<tr><th>Points plotted</th><th>P</th><th>X co-ordinate</th><th>Y co-ordinate</th></tr>
<tr><td>1</td><td></td><td>1</td><td>1</td></tr>
<tr><td>2</td><td>-5</td><td>2</td><td>2</td></tr>
<tr><td>3</td><td>3</td><td>3</td><td>2</td></tr>
<tr><td>4</td><td>-3</td><td>4</td><td>3</td></tr>
<tr><td>5</td><td>5</td><td>5</td><td>3</td></tr>
<tr><td>6</td><td>-1</td><td>6</td><td>4</td></tr>
<tr><td>7</td><td>7</td><td>7</td><td>4</td></tr>
<tr><td>8</td><td>1</td><td>8</td><td>5</td></tr>
</table>
</td><td>Table correctly filled in.</td><td></td><td></td></tr>
<tr><td>(iii)</td><td></td><td>Correct line plotted.</td><td></td><td></td></tr>
</table>

Question	Sample evidence	Achievement	Achievement with Merit	Achievement with Excellence
(iv)	Gray in surrounding edges to trick the eye into seeing less jagged edges, or use a retina display with more smaller pixels. 	Correct answer, explains anti-aliasing, uses shading to fill pixels to smooth line.		
(d)	There are lots of calculations that are faster on a GPU. Identifies kinds of repetitive calculations and what a GPU does to help.	Gives a simple (correct) answer.	Gives repetitive calculations.	Examples of other computer graphics and how they are or could be used. Any of: • innovative and imaginative connections • exploration of less obvious implications • making justified predictions • suggesting improvements • making justified generalisations that could be applied beyond the area itself • use of higher-level thinking skills such as synthesis.

Computer Vision

Question	Sample evidence	Achievement	Achievement with Merit	Achievement with Excellence
THREE (a) (i)	Noise often appears as random changes to pixels which can make it harder to recognise the makeup of an image and detect edges. Noise reduction is important, but you have to be careful not to discard useful information. Noise reduction techniques have to predict which pixels are supposed to be there and which aren't.	Identifies that noise in an image makes detecting edges more difficult.		
(ii)	Blurring is often the first step in reducing noise in an image. Applying a Gaussian Blur smooths the edges, and while single pixels will be blurred, the stronger edge signal remains.	Explains how an identified issue is addressed.		
(b) (i)	Canny edge detection characterises the edges of an image, allowing for its identification.	Explains process of finding edges and explains why this is important.		
(ii)	Canny edge detection is a simple but accurate method for the edge detection problem, with more demanding requirements on the accuracy and robustness of the detection. The Canny edge detection algorithm has two fixed global threshold values to remove the false edges. While it works well for some images, its results can vary.	Explains how a Canny edge detector is used. Other valid responses are acceptable.		
(iii)	A Gaussian filter which is applied to reduce noise can also smooth edges, but could miss weak edges. One challenge involved in implementing accurate Canny edge detection is that as an image increases in complexity, different threshold values need to be calculated to find actual edges. When global thresholds are calculated manually, issues with the ability to scale and resize the image arise.	Identifies and explains challenges.	Offers solutions to challenges.	
(iv)	Canny edge detection provides a simple but precise method for the edge detection problem. If more accuracy and robustness is required the traditional algorithm may no longer function. SOBEL does not place any emphasis on pixels that are closer to the centre of the masks. Applying a low-pass blurring filter (such as a Gaussian Blur) smooths edges and removes noise from an image.		Provides solutions to challenges.	Provides solutions to challenges with clear links to computer science algorithms.
(c)	3D effect, triangulation, and differences between camera can detect angle and extrapolate to depth. Increasing distance can create a problem when trying to calculate depth accurately. Can explain a variety of techniques, e.g. stereo, Lidar, etc.	Identifies and explains concept.		

Question	Sample evidence	Achievement	Achievement with Merit	Achievement with Excellence
(d) (i)	They use distinguishable landmarks, nodal points, peaks and valleys. For example, the distance between the eyes, width of the nose, depth of the eye sockets, shape of the cheekbones, length of the jawline. 2D facial recognition uses other 2D images to compare the image against. It requires the subject be looking directly at the camera and for there to be little difference from the image in the database. 3D facial recognition depends on depth, different angles, and distinctive features of the face (rigid tissue and bone such as eye sockets, nose, and chin) which don't change over time. It uses a series of steps to verify identity. The information is translated into a set of numbers to represent the features of a subject's face.	Identifies facial recognition makeup.	Explains and links to examples.	Other examples of computer vision and how they are or could be used. Any of: • innovative and imaginative connections • exploration of less obvious implications • making justified predictions • suggesting improvements • making justified generalisations that could be applied beyond the area itself • use of higher-level thinking skills such as synthesis.
(ii)	Face detection is a broader term than face recognition, and means the system can identify that a human face is present. Facial recognition can confirm identity, which is why it is used to control access to sensitive areas.	Identifies differences.	Explains differences and links to examples.	
(iii)	Facial recognition is a biometric technology and can be used to unlock phones and specific applications, which can help to save people time. It is also used in surveillance and by the authorities to solve crimes.			Refers to positive and negative impacts and can justify explanations.
(e)	• Self-driving cars use computer vision to make sense of their surroundings. Cameras on the cars capture video and process the images in real-time to gather information to make decisions. • Facial recognition is used to authenticate, detect, and tag users. • Augmented and mixed reality overlay and embed virtual objects on real-world imagery. • Computer vision is used in healthcare to automate tasks such as detecting disease and finding symptoms on medical scans.			Identifies changes in computer vision and links to computer science concepts from example Explains and links to examples. Links are with increased capability in processing, detection and recognition, with more applications. Links to broader computer science concepts. Other examples of computer vision and how they are or could be used. • innovative and imaginative connections • exploration of less obvious implications • making justified predictions • suggesting improvements

Question	Sample evidence	Achievement	Achievement with Merit	Achievement with Excellence
				• making justified generalisations that could be applied beyond the area itself • use of higher-level thinking skills such as synthesis.