

**Cambridge International Examinations** 

Cambridge International General Certificate of Secondary Education

## **COMPUTER SCIENCE**

0478/12

Paper 1 Computer Systems

February/March 2023

MARK SCHEME

Maximum Mark : 75



[Turn over

### PRE-STANDARDISATION Cambridge Assessment International Education – Generic Marking Principles

These general marking principles must be applied by all examiners when marking candidate answers. They should be applied alongside the specific content of the mark scheme or generic level descriptors for a question. Each question paper and mark scheme will also comply with these marking principles.

GENERIC MARKING PRINCIPLE 1:

Marks must be awarded in line with:

- the specific content of the mark scheme or the generic level descriptors for the question
- the specific skills defined in the mark scheme or in the generic level descriptors for the question
- the standard of response required by a candidate as exemplified by the standardisation scripts.

GENERIC MARKING PRINCIPLE 2:

Marks awarded are always whole marks (not half marks, or other fractions).

GENERIC MARKING PRINCIPLE 3:

Marks must be awarded **positively**:

- marks are awarded for correct/valid answers, as defined in the mark scheme. However, credit is given for valid answers which go beyond the scope of the syllabus and mark scheme, referring to your Team Leader as appropriate
- marks are awarded when candidates clearly demonstrate what they know and can do
- marks are not deducted for errors
- marks are not deducted for omissions
- answers should only be judged on the quality of spelling, punctuation and grammar when these features are specifically assessed by the question as indicated by the mark scheme. The meaning, however, should be unambiguous.

GENERIC MARKING PRINCIPLE 4:

Rules must be applied consistently e.g. in situations where candidates have not followed instructions or in the application of generic level descriptors.

GENERIC MARKING PRINCIPLE 5:

Marks should be awarded using the full range of marks defined in the mark scheme for the question (however; the use of the full mark range may be limited according to the quality of the candidate responses seen).

**GENERIC MARKING PRINCIPLE 6:** 

Marks awarded are based solely on the requirements as defined in the mark scheme. Marks should not be awarded with grade thresholds or grade descriptors in mind.

Question	Answer	Marks	Guidance
1(a)	D	1	
1(b)	<ul> <li>One mark for identification. E.g. One mark per bullet for description to max two each.</li> <li>Virus <ul> <li>Software that replicates</li> <li>Deletes/damages data/hardware // equivalent</li> </ul> </li> </ul>	6	
	<ul> <li>Worm</li> <li>Software that replicates itself on a network</li> <li>Takes-up bandwidth/storage space</li> </ul>		
	<ul> <li>Trojan horse</li> <li>Software that is hidden within other software // Software that is disguised as authentic software</li> <li>Deletes/damages data/hardware</li> </ul>		
	<ul> <li>Adware</li> <li>Software that generates/displays adverts on a user's computer</li> <li>Some may contain spyware</li> <li>Some when clicked may link to viruses</li> </ul>		
	<ul> <li>Ransomware</li> <li>Software that stops a user accessing/using their computer/data</li> <li>A fee has to be paid to 'release' the computer/device</li> <li>otherwise the data will be deleted</li> </ul>		

1(c)       One mark for each similarity to max two. One mark for difference (both sides needed unless implied)       3         Similarities e.g.       Encek incoming and outgoing signals       5         Store white/black list // by example       Block access to incoming signals       6         Both block unauthorised access//by example       6       Block access//by example         Keep a log of traffic       0       0         Differences e.g.       Proxy intention is to divert attack from server, firewall is to stop unauthorised access       9         Proxy protects a server/network, firewall potects network/individual computer       9       9         Proxy protects a server/network, firewall potects network/individual computer       4       Allow in reverse e.g. justifying high-level         2(a)       No mark for choice.       1       1       1         four from matching choice.       High-level       9       2       1         Machine independent       therefore, the programmer is less likely to make mistakes // can write in shorter timeframe       2       2       Choice must be clear, if not written in space then read answer and if not obvious which is chosen, 0 marks.         Low-level       the game can be used on any computer without a need for understanding of the hardware       2       Choice must be clear, if not written in space then read answer and if not obvious which is chosen, 0 marks.       2<				
Similarities e.g.       Check incoming and outgoing signals       Image: Store white/black list // by example	1(c)	One mark for each similarity to max two. One mark for difference (both sides needed unless implied)	3	
<ul> <li>Check incoming and outgoing signals</li> <li>Store white/black list // by example</li> <li>Block access to incoming signals</li> <li>Both block unauthorised access/by example</li> <li>Keep a log of traffic</li> <li>Differences e.g.</li> <li>Proxy intention is to divert attack from server, firewall is to stop unauthorised accesss</li> <li>Proxy protects a server/network, firewall protects network/individual computer</li> <li>Proxy filters access from items such as websites, firewall blocks ports</li> <li>2(a) No mark for choice.</li> <li>four from matching choice.</li> <li>High-level</li> <li>Easier/faster to read/write/understand</li> <li> therefore, the programmer is less likely to make mistakes // can write in shorter timeframe</li> <li>Easier to debug</li> <li>so the programmer can find and correct errors in less time</li> <li>Machine independent</li> <li>the game can be used on any computer without a need for understanding of the hardware</li> <li>Low-level</li> <li>More memory efficient</li> <li> computer game may require high memory consumption so this needs limiting</li> <li>Game may be being built for one device</li> </ul>		Similarities e.g.		
<ul> <li>Store white/black list // by example</li> <li>Block access to incoming signals</li> <li>Both block unauthorised access/by example</li> <li>Keep a log of traffic</li> <li>Differences e.g.</li> <li>Proxy can hide user's IP address, firewall does not</li> <li>Proxy protects a server/network, firewall protects network/individual computer</li> <li>Proxy protects a server/network, firewall protects network/individual computer</li> <li>Proxy filters access from items such as websites, firewall blocks ports</li> <li>2(a) No mark for choice.</li> <li>four from matching choice.</li> <li>High-level</li> <li>Easiert/faster to read/write/understand</li> <li> therefore, the programmer is less likely to make mistakes // can write in shorter timeframe</li> <li>Easier to debug</li> <li>so the programmer can find and correct errors in less time</li> <li>Machine independent</li> <li>the game can be used on any computer without a need for understanding of the hardware</li> <li>Low-level</li> <li>More memory efficient</li> <li> computer game may require high memory consumption so this needs limiting</li> <li>Game may be being built for one device</li> </ul>		Check incoming and outgoing signals		
<ul> <li>Block access to incoming signals</li> <li>Both block unauthorised access//by example</li> <li>Keep a log of traffic</li> <li>Differences e.g.</li> <li>Proxy can hide user's IP address, firewall does not</li> <li>Proxy intention is to divert attack from server, firewall is to stop unauthorised access</li> <li>Proxy protects a server/network, firewall protects network/individual computer</li> <li>Proxy fitters access from items such as websites, firewall blocks ports</li> <li>2(a) No mark for choice.</li> <li>High-level</li> <li>Easier/faster to read/write/understand</li> <li> therefore, the programmer is less likely to make mistakes // can write in shorter timeframe</li> <li>Easier to debug</li> <li>so the programmer can find and correct errors in less time</li> <li>Machine independent</li> <li>the game can be used on any computer without a need for understanding of the hardware</li> <li>Low-level</li> <li>More memory efficient</li> <li>computer game may require high memory consumption so this needs limiting</li> <li>Game may be being built for one device</li> </ul>		Store white/black list // by example		
<ul> <li>Both block unauthorised access//by example</li> <li>Keep a log of traffic</li> <li>Differences e.g.</li> <li>Proxy can hide user's IP address, firewall does not</li> <li>Proxy intention is to divert attack from server, firewall is to stop unauthorised access</li> <li>Proxy protects a server/network, firewall protects network/individual computer</li> <li>Proxy filters access from items such as websites, firewall blocks ports</li> <li>2(a) No mark for choice.</li> <li>four from matching choice.</li> <li>High-level</li> <li>Easier/faster to read/write/understand</li> <li> therefore, the programmer is less likely to make mistakes // can write in shorter timeframe</li> <li>Easier to debug</li> <li> so the programmer can find and correct errors in less time</li> <li>Machine independent</li> <li> tothe game can be used on any computer without a need for understanding of the hardware</li> <li>Low-level</li> <li>More memory efficient</li> <li> computer game may require high memory consumption so this needs limiting</li> <li>Game may be being built for one device</li> </ul>		Block access to incoming signals		
<ul> <li>Keep a log of traffic</li> <li>Differences e.g.</li> <li>Proxy can hide user's IP address, firewall does not</li> <li>Proxy protects a server/network, firewall is to stop unauthorised access</li> <li>Proxy protects a server/network, firewall protects network/individual computer</li> <li>Proxy filters access from items such as websites, firewall blocks ports</li> <li>Xo mark for choice.</li> <li>four from matching choice.</li> <li>High-level</li> <li>Easier/faster to read/write/understand</li> <li> therefore, the programmer is less likely to make mistakes // can write in shorter timeframe</li> <li>Easier to debug</li> <li>so the programmer can find and correct errors in less time</li> <li>Machine independent</li> <li>so the programmer can find and correct errors in less time</li> <li>Machine independent</li> <li>so the programmer can find and correct errors in less time</li> <li>Machine independent</li> <li>so the programmer can find and correct errors in less time</li> <li>Machine independent</li> <li>computer game may require high memory consumption so this needs limiting</li> <li>Game may be being built for one device</li> </ul>		Both block unauthorised access//by example		
Differences e.g.       • Proxy can hide user's IP address, firewall does not       • Proxy intention is to divert attack from server, firewall is to stop unauthorised access       • Proxy intention is to divert attack from server, firewall is to stop unauthorised access       • Proxy protects a server/network, firewall protects network/individual computer         2(a)       No mark for choice.       4       Allow in reverse e.g. justifying high-level by saying why low-level by saying why low-level should not be used.         2(a)       No mark for choice.       4       Allow in reverse e.g. justifying high-level by saying why low-level should not be used.         2(a)       No mark for choice.       4       Allow in reverse e.g. justifying high-level by saying why low-level should not be used.         • Easier/faster to read/write/understand       • therefore, the programmer is less likely to make mistakes // can write in shorter timeframe       • Ecasier to debug         • so the programmer can find and correct errors in less time       • Choice must be clear, if not written in space then read answer and if not obvious which is chosen, 0 marks.         Low-level       • Machine independent       • computer game may require high memory consumption so this needs limiting       • Game may be being built for one device		Keep a log of traffic		
Differences e.g. • Proxy can hide user's IP address, firewall does not • Proxy intention is to divert attack from server, firewall is to stop unauthorised access • Proxy protects a server/network, firewall protects network/individual computer • Proxy filters access from items such as websites, firewall blocks ports4Allow in reverse e.g. justifying high-level by saying why low- level should not be used.2(a)No mark for choice. four from matching choice. High-level • Easier/faster to read/write/understand • therefore, the programmer is less likely to make mistakes // can write in shorter timeframe • Easier to debug • so the programmer can find and correct errors in less time • Machine independent • the game can be used on any computer without a need for understanding of the hardwareChoice must be clear, if not written in space then read answer and if not obvious which is chosen, 0 marks.Low-level • More memory efficient • computer game may require high memory consumption so this needs limiting • Game may be being built for one deviceImage: Sail to sail				
<ul> <li>Proxy can hide user's IP address, firewall does not</li> <li>Proxy intention is to divert attack from server, firewall is to stop unauthorised access</li> <li>Proxy protects a server/network, firewall protects network/individual computer</li> <li>Proxy filters access from items such as websites, firewall blocks ports</li> <li>Allow in reverse e.g., justifying high-level</li> <li>by saying why low-level</li> <li>Easier/faster to read/write/understand</li> <li> therefore, the programmer is less likely to make mistakes // can write in shorter timeframe</li> <li>Easier to debug</li> <li>so the programmer can find and correct errors in less time</li> <li>Machine independent</li> <li>the game can be used on any computer without a need for understanding of the hardware</li> <li>Low-level</li> <li>More memory efficient</li> <li> computer game may require high memory consumption so this needs limiting</li> <li>Game may be being built for one device</li> </ul>		Differences e.g.		
<ul> <li>Proxy intention is to divert attack from server, firewall is to stop unauthorised access</li> <li>Proxy protects a server/network, firewall protects network/individual computer</li> <li>Proxy filters access from items such as websites, firewall blocks ports</li> <li>2(a) No mark for choice.</li> <li>four from matching choice.</li> <li>High-level</li> <li>Easier/faster to read/write/understand</li> <li> therefore, the programmer is less likely to make mistakes // can write in shorter timeframe</li> <li>Easier to debug</li> <li> so the programmer can find and correct errors in less time</li> <li>Machine independent</li> <li> the game can be used on any computer without a need for understanding of the hardware</li> <li>Low-level</li> <li>More memory efficient</li> <li> computer game may require high memory consumption so this needs limiting</li> <li>Game may be being built for one device</li> </ul>		<ul> <li>Proxy can hide user's IP address, firewall does not</li> </ul>		
<ul> <li>Proxy protects a server/network, firewall protects network/individual computer</li> <li>Proxy filters access from items such as websites, firewall blocks ports</li> <li>Allow in reverse e.g. justifying high-level by saying why low-level</li> <li>Easier/faster to read/write/understand</li> <li> therefore, the programmer is less likely to make mistakes // can write in shorter timeframe</li> <li>Easier to debug</li> <li> so the programmer can find and correct errors in less time</li> <li>Machine independent</li> <li> the game can be used on any computer without a need for understanding of the hardware</li> <li>Low-level</li> <li>More memory efficient</li> <li> computer game may require high memory consumption so this needs limiting</li> <li>Game may be being built for one device</li> </ul>		<ul> <li>Proxy intention is to divert attack from server, firewall is to stop unauthorised access</li> </ul>		
<ul> <li>Proxy filters access from items such as websites, firewall blocks ports</li> <li>2(a) No mark for choice.</li> <li>four from matching choice.</li> <li>High-level</li> <li>Easier/faster to read/write/understand</li> <li> therefore, the programmer is less likely to make mistakes // can write in shorter timeframe</li> <li>Easier to debug</li> <li>so the programmer can find and correct errors in less time</li> <li>Machine independent</li> <li>the game can be used on any computer without a need for understanding of the hardware</li> <li>Low-level</li> <li>More memory efficient</li> <li> computer game may require high memory consumption so this needs limiting</li> <li>Game may be being built for one device</li> </ul>		<ul> <li>Proxy protects a server/network, firewall protects network/individual computer</li> </ul>		
2(a)       No mark for choice.       4       Allow in reverse e.g. justifying high-level by saying why low-level should not be used.         4       Allow in reverse e.g. justifying high-level by saying why low-level should not be used.         •       Easier/faster to read/write/understand       by saying why low-level should not be used.         •       Easier to debug       Choice must be clear, if not written in space then read answer and if not obvious which is chosen, 0 marks.         •       Machine independent       Space then read answer and if not obvious which is chosen, 0 marks.         •       Low-level       More memory efficient         •       computer game may require high memory consumption so this needs limiting       Game may be being built for one device		<ul> <li>Proxy filters access from items such as websites, firewall blocks ports</li> </ul>		
four from matching choice.       justifying high-level         High-level       justifying high-level         • Easier/faster to read/write/understand       level should not be         • therefore, the programmer is less likely to make mistakes // can write in shorter       level should not be         • Easier to debug       Choice must be         •so the programmer can find and correct errors in less time       Choice must be         • Machine independent       space then read         •the game can be used on any computer without a need for understanding of the hardware       answer and if not obvious which is chosen, 0 marks.         Low-level       More memory efficient       computer game may require high memory consumption so this needs limiting         • Game may be being built for one device       Game may be being built for one device	2(a)	No mark for choice.	4	Allow in reverse e.a.
High-level       by saying why low-level should not be         • Easier/faster to read/write/understand       by saying why low-level should not be         • therefore, the programmer is less likely to make mistakes // can write in shorter       timeframe         • Easier to debug       Choice must be         •so the programmer can find and correct errors in less time       Choice must be         • Machine independent       space then read         •the game can be used on any computer without a need for understanding of the       answer and if not         • hardware       by obvious which is         Low-level       More memory efficient         • More memory efficient       computer game may require high memory consumption so this needs limiting         • Game may be being built for one device       High-level	(/	four from matching choice.		iustifving high-level
<ul> <li>Easier/faster to read/write/understand</li> <li> therefore, the programmer is less likely to make mistakes // can write in shorter timeframe</li> <li>Easier to debug</li> <li> so the programmer can find and correct errors in less time</li> <li>Machine independent</li> <li> the game can be used on any computer without a need for understanding of the hardware</li> <li>Low-level</li> <li>More memory efficient</li> <li> computer game may require high memory consumption so this needs limiting</li> <li>Game may be being built for one device</li> </ul>		High-level		by saying why low-
<ul> <li> therefore, the programmer is less likely to make mistakes // can write in shorter timeframe</li> <li>Easier to debug</li> <li>so the programmer can find and correct errors in less time</li> <li>Machine independent</li> <li>the game can be used on any computer without a need for understanding of the hardware</li> <li>Low-level</li> <li>More memory efficient</li> <li>computer game may require high memory consumption so this needs limiting</li> <li>Game may be being built for one device</li> </ul>		<ul> <li>Easier/faster to read/write/understand</li> </ul>		level should not be
<ul> <li>timeframe</li> <li>Easier to debug</li> <li>so the programmer can find and correct errors in less time</li> <li>Machine independent</li> <li>the game can be used on any computer without a need for understanding of the hardware</li> <li>Low-level</li> <li>More memory efficient</li> <li> computer game may require high memory consumption so this needs limiting</li> <li>Game may be being built for one device</li> </ul>		• therefore, the programmer is less likely to make mistakes // can write in shorter		used.
<ul> <li>Easier to debug</li> <li>so the programmer can find and correct errors in less time</li> <li>Machine independent</li> <li>the game can be used on any computer without a need for understanding of the hardware</li> <li>Low-level</li> <li>More memory efficient</li> <li> computer game may require high memory consumption so this needs limiting</li> <li>Game may be being built for one device</li> </ul>		timeframe		
<ul> <li>so the programmer can find and correct errors in less time</li> <li>Machine independent</li> <li>the game can be used on any computer without a need for understanding of the hardware</li> <li>Low-level</li> <li>More memory efficient</li> <li> computer game may require high memory consumption so this needs limiting</li> <li>Game may be being built for one device</li> </ul>		Easier to debug		Choice must be
<ul> <li>Machine independent</li> <li>the game can be used on any computer without a need for understanding of the hardware</li> <li>Low-level</li> <li>More memory efficient</li> <li> computer game may require high memory consumption so this needs limiting</li> <li>Game may be being built for one device</li> </ul>		<ul> <li>so the programmer can find and correct errors in less time</li> </ul>		clear, if not written in
<ul> <li>the game can be used on any computer without a need for understanding of the hardware</li> <li>Low-level</li> <li>More memory efficient</li> <li> computer game may require high memory consumption so this needs limiting</li> <li>Game may be being built for one device</li> </ul>		Machine independent		space then read
hardware       obvious which is chosen, 0 marks.         Low-level       More memory efficient         • More memory efficient       computer game may require high memory consumption so this needs limiting         • Game may be being built for one device		<ul> <li>the game can be used on any computer without a need for understanding of the</li> </ul>		answer and if not
Low-level       chosen, 0 marks.         • More memory efficient          • computer game may require high memory consumption so this needs limiting          • Game may be being built for one device		hardware		obvious which is
<ul> <li>Low-level</li> <li>More memory efficient</li> <li> computer game may require high memory consumption so this needs limiting</li> <li>Game may be being built for one device</li> </ul>				chosen, 0 marks.
<ul> <li>More memory efficient</li> <li> computer game may require high memory consumption so this needs limiting</li> <li>Game may be being built for one device</li> </ul>		Low-level		
<ul> <li> computer game may require high memory consumption so this needs limiting</li> <li>Game may be being built for one device</li> </ul>		More memory efficient		
Game may be being built for one device		computer game may require high memory consumption so this needs limiting		
		Game may be being built for one device		

0478/12	2
---------	---

i			
	<ul> <li> therefore, machine independence is not required</li> <li>Direct manipulation of hardware</li> <li> allows for more efficient control/response time</li> </ul>		
2(b)	<ul> <li>two from for each compiler and interpreter</li> <li>Compiler <ul> <li>Checks all code at once before translating/executing any code</li> <li>Produces error report with all errors found for the whole code</li> <li>Produces executable file</li> </ul> </li> </ul>	4	
	<ul> <li>Interpreter</li> <li>Checks one line of code and then translates/executes it</li> <li>Stops when an error is found</li> <li> when corrected the program can be run from the same position</li> </ul>		
3(a)	<b>One</b> mark each: 8 bytes = <b>16</b> nibbles 512 KiB = <b>0.5</b> MiB 4 GiB = <b>4096</b> MiB 1 EiB = <b>1024</b> PiB	4	
3(b)	<ul> <li>Any three from:</li> <li>Currently running data</li> <li>Currently running software</li> <li>Currently running instructions</li> <li>Currently running parts of OS</li> </ul>	3	Consider FT missing currently running
3(c)	<ul> <li>Any two from:</li> <li>For non-volatile/permanent/long-term storage of files/data</li> <li>To store data that is not currently required by the CPU</li> <li>To store data to transfer it to another computer</li> </ul>	2	
4(a)	One mark each:	2	

	TRE-STANDARDISATION		
	<ul> <li>Pixel: <ul> <li>One square/circle of one colour // the smallest component of the image</li> </ul> </li> <li>Resolution: <ul> <li>The number of pixels per set area/cm/inch // the number of pixels wide by the number of pixels high // number of pixels in an image 'dpi' on its own is NE</li> </ul> </li> </ul>		
4(b)	One mark for answer: • 2,000,000 bytes One mark for working from: • 1000 x 1000 (= 1,000,000) • 1,000,000 * 2 (= 2,000,000)	2	
4(c)	One mark for each term:         Solid-state storage is non-volatile, this means that the data is not lost when the power is turned off.         Solid-state storage is made of transistors that are laid out in a grid.         Gates are used to control the flow of the electrons through the transistors. This changes the data in the transistors from 1 to 0, or from 0 to 1.	4	
4(d)	<ul> <li>Any one from:</li> <li>Reduce the file size</li> <li>Increase transmission speed // Reduce transmission time</li> <li>Reduce storage space required</li> <li>Less bandwidth required for transmission</li> </ul>	1	
5(a)(i)	Any <b>two</b> from e.g. • Destination address	2	

0478/12

	FRE-STANDARDISATION		
	<ul> <li>Packet number</li> <li>Originators address</li> <li>Error detection method</li> </ul>		
5(a)(ii)	One mark each: <ul> <li>Payload</li> <li>Trailer</li> </ul>	2	Consider other features not in header
5(b)(i)	No mark for choice Three from for justification that matches choice Serial • Data arrives in order sent • less likely to have errors • Can transmit over a longer distance (i.e. another room) • Still fast transmission •sufficient for this purpose Parallel • Faster transmission speed than serial • faster response to requests • Very long connection not needed	3	
5(b)(ii)	next room is (likely) within distance for parallel	2	
5(0)(11)	<ul> <li>Data goes in both directions</li> <li>at the same time</li> </ul>	L	

0478/12

					1 1/1						
5(c)(i)	i) <b>One</b> mark each:								2		
		0	1	1	0	0	0	1	1		
		0	0	0	0	0	0	0	0		
5(c)(ii)	One mark each In parity The pos	h: y check, i sition of b	nterchang ooth chang	e of bits w ges will be	vill not be o highlighte	detected				2	
5(c)(iii)	<ul> <li>&gt;)(iii) One mark each:</li> <li>Bit 6</li> <li>Byte 4</li> </ul>									2	
5(d)(i)	<ul> <li>Any two from:</li> <li>e.g.</li> <li>Storing bookmarks/favourites</li> <li>Storing history</li> <li>Allow multiple tabs to be open</li> <li>Allow movement back and forth between web pages // provides navigation tools</li> <li>Address bar</li> </ul>								2		
5(d)(ii)	<ul> <li>Address bar</li> <li>Any two from:</li> <li>e.g.</li> <li>Storing preferences</li> <li>Storing account details</li> <li>Storing recent purchases</li> <li>To display relevant adverts</li> </ul>										

	Storing shopping	ng basket				
6(a)	One mark each	4				
	Movement	Binary	Denary	Hexadecimal		
	forward 1 step	00011111	31	1F		
	back 1 step	10001100	140	8C		
	turn right	01011010	90	5A		
	turn left	(0)1111000	120	78		
6(b)	Any <b>two</b> from: • The design of revealed the construction • The operation of the operatio	2				
6(c)	Seven from: Uses an infra-re Sensor continu Microprocessor If the data/signa a signal is se If the data/signa a signal is se The whole proc	7				

0478/12

i			
6(d)(i)	Any three from:	3	
	Collecting data		
	Stores rules for using the data		
	The ability to reason		
	The ability to learn		
	The ability to adapt		
	The ability to change its own rules		
	The ability to change its own data		
6(d)(ii)	Four from e.g.:	4	
	Use machine learning algorithms …		
	to adapt its behaviour/rules/data		
	Collects data about where it has been		
	to make sure it does not repeat the same route		
	Collect data about common obstacles/problems		
	so, it knows how to react next time		
	Store successful actions		
	<ul> <li>so, it knows what is most likely to work next time</li> </ul>		