Cambridge Assessment International Education
Cambridge International General Certificate of Secondary Education

COMPUTER SCIENCE **0478/12**

Paper 1 **October/November 2018**

MARK SCHEME

Maximum Mark: 75

Published

This mark scheme is published as an aid to teachers and candidates, to indicate the requirements of the examination. It shows the basis on which Examiners were instructed to award marks. It does not indicate the details of the discussions that took place at an Examiners' meeting before marking began, which would have considered the acceptability of alternative answers.

Mark schemes should be read in conjunction with the question paper and the Principal Examiner Report for Teachers.

Cambridge International will not enter into discussions about these mark schemes.

Cambridge International is publishing the mark schemes for the October/November 2018 series for most Cambridge IGCSE™, Cambridge International A and AS Level components and some Cambridge O Level components.

This syllabus is approved for use in England, Wales and Northern Ireland as a Cambridge International Level 1/Level 2 Certificate.

This document consists of **12** printed pages.

PUBLISHED

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.

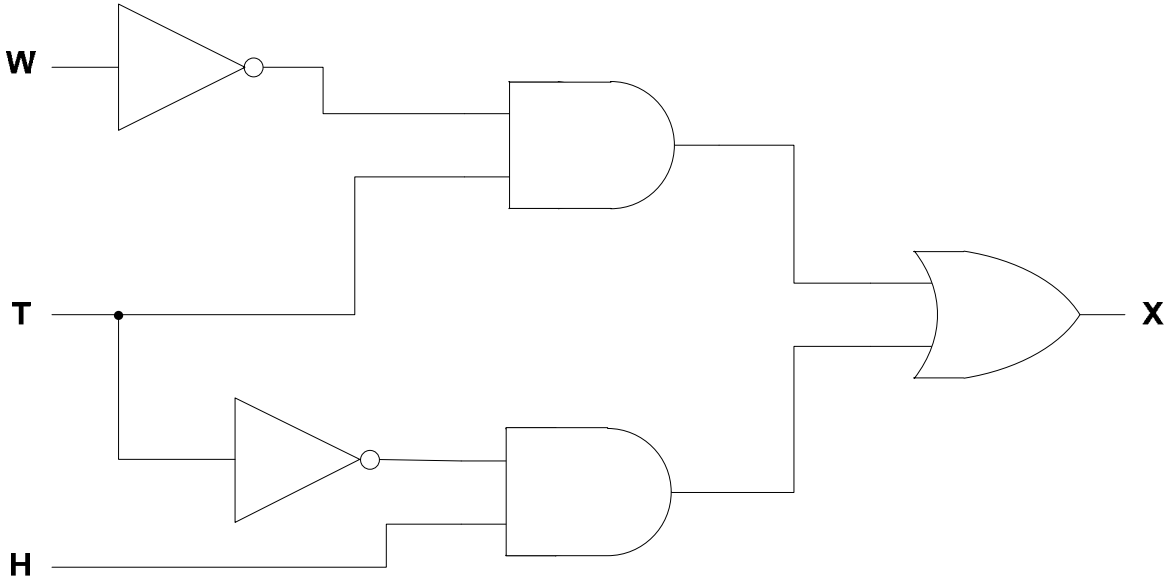
GENERIC MARKING PRINCIPLE 4:

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
1(a)	1 mark for each correct 8-bit binary number 66 | 0 | 1 | 0 | 0 | 0 | 0 | 1 | 0 | 85 | 0 | 1 | 0 | 1 | 0 | 1 | 0 | 1 | 83 | 0 | 1 | 0 | 1 | 0 | 0 | 1 | 1 |	**3**
1(b)(i)	1 mark for each correct hexadecimal number 4B 45 59	**3**
1(b)(ii)	**Three** from: • (HTML) colour codes • Error messages • MAC addresses • IP addresses • Assembly language • Memory dump • Locations in memory	**3**
1(b)(iii)	**Two** from: • Easier to read/write/understand (for humans) • Easier to remember (for humans) • Short way to represent binary // Uses less **screen/display** space • Fewer errors made (in data transcription) • Easier to debug (for humans)	**2**

<table>
<tr><th>Question</th><th>Answer</th><th>Marks</th></tr>
<tr><td>2(a)</td><td>1 mark for each correct tick (✓)
<table>
<tr><th>Statement</th><th>RAM (✓)</th><th>ROM (✓)</th></tr>
<tr><td>Stores the programs and data that are currently in use</td><td>✓</td><td></td></tr>
<tr><td>Used to boot up the computer when power is turned on</td><td></td><td>✓</td></tr>
<tr><td>Contents are retained when power is turned off</td><td></td><td>✓</td></tr>
</table></td><td>3</td></tr>
<tr><td>2(b)</td><td>Primary</td><td>1</td></tr>
<tr><td>2(c)</td><td>Two from:
• Non-volatile storage
• Storage that can be disconnected/removed from the computer
• Any suitable example
• Must be (physically) connected to computer to obtain stored data
• Used to store files as a backup</td><td>2</td></tr>
</table>

Question	Answer	Marks
3	1 mark for each correct logic gate, with correct inputs. 	5

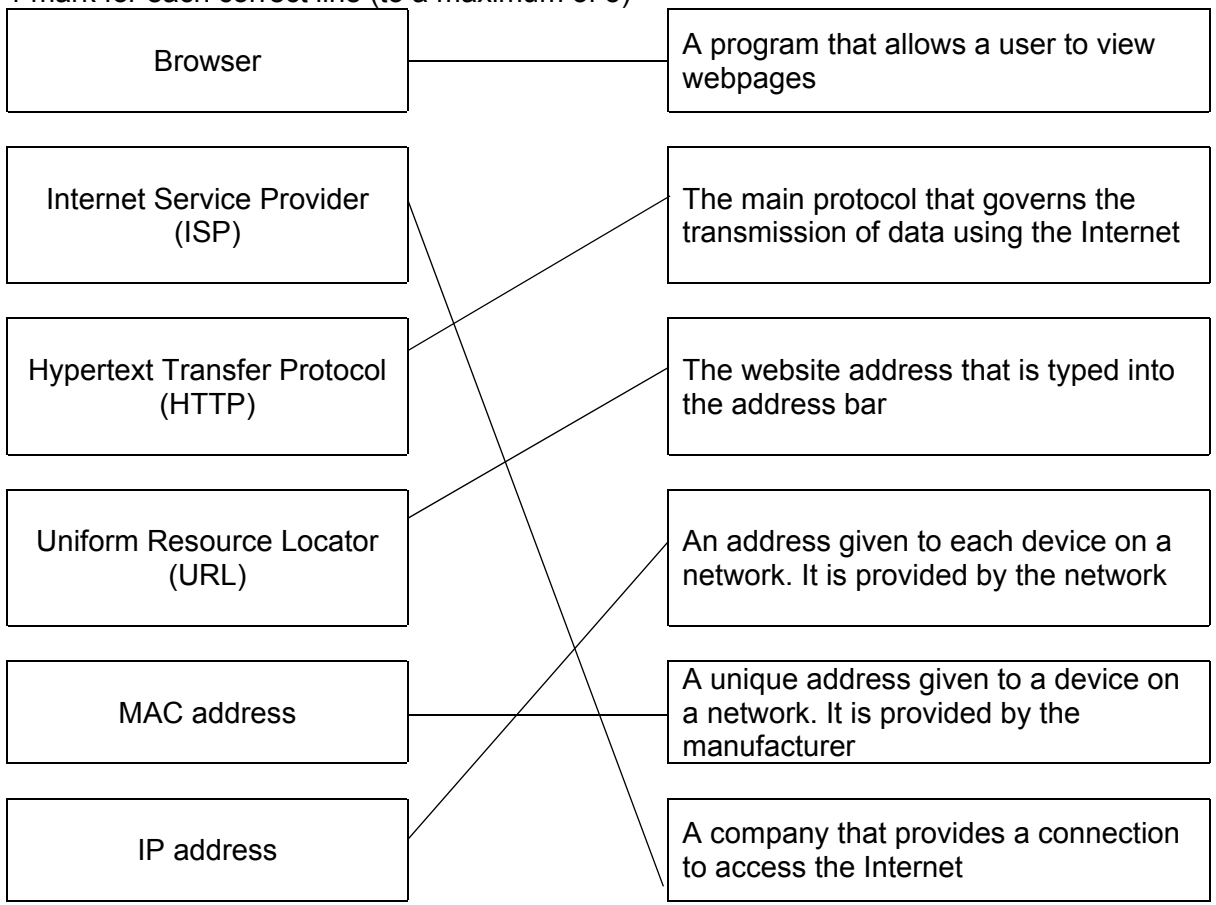
Question	Answer	Marks
4(a)	**Three** from: • Malware • Virus // No antivirus • Denial of service • Spyware // No antispyware • Phishing // opening unknown links/emails • Pharming // opening unknown links/emails (only award once for this alternative) • Hacking/cracking/unauthorised access // No/weak password // No/weak firewall • Downloading/Using unknown software • Not updating software • Physical issue e.g. computer/door left unlocked	**3**
4(b)	**Four** from: • It examines/monitors/filters traffic into **and** out of a computer • It allows a user to set criteria/rules for the traffic • It checks whether the traffic meets the criteria/rules • It blocks any traffic that does not meet the criteria/rules // Blocks unauthorised access • It warns a **user** of any unauthorised software/access/unauthorised outgoing traffic • It keeps a log of all traffic (that can be examined)	**4**

Question	Answer	Marks
5(a)(i)	2D/3D cutter	**1**
5(a)(ii)	Liquid crystal display // LCD	**1**
5(a)(iii)	Actuator	**1**
5(b)	1 mark for each correct missing word, in the given order: • interactive whiteboard • inkjet • thermal bubble • laser • rotating	**5**

Question	Answer	Marks
6(a)	• Compiler • Interpreter	**2**
6(b)	**Four** from: • Closer to human language/English … • … so it is easier/quicker to read/write/understand • … so it is easier/quicker to debug the program • … therefore, less likely to make errors • The program can be used on many different platforms … • … because it is written in source code • … because it is compiled into object code • They have built-in functions/libraries … • … this saves time when writing the program • Do not need to manipulate memory addresses directly … • … therefore, specialist knowledge of this is not required • Only need to learn a single language … • … as this can be used on many different computers	**4**

<table>
<tr><th>Question</th><th>Answer</th><th>Marks</th></tr>
<tr><td>6(c)</td><td>1 mark for each correct tick (✓)
<table>
<tr><th>Computer code</th><th>High-level language (✓)</th><th>Assembly language (✓)</th><th>Machine code (✓)</th></tr>
<tr><td>10110111
11001100
01011100</td><td></td><td></td><td>✓</td></tr>
<tr><td>FOR X = 1 TO 10
PRINT X
NEXT X</td><td>✓</td><td></td><td></td></tr>
<tr><td>INP X
STA X
LDA Y</td><td></td><td>✓</td><td></td></tr>
</table></td><td>3</td></tr>
</table>

Question	Answer	Marks
7	1 mark for each correct line (to a maximum of 5)	5



Question	Answer	Marks
8	**Four** from: • Used to attend to certain tasks/issues • Used to make sure that **vital** tasks are dealt with **immediately** • The interrupt/signal tells the CPU/processor (that its attention is required) • A signal that can be sent from a device (attached to the computer) • A signal that can be sent from software (installed on the computer) • The interrupt will cause the OS/current process to pause • The OS/CPU/ISR will service/handle the interrupt • They have different levels of priority • After the interrupt is serviced, the (previous) process is continued • It enables multi-tasking to be carried out on a computer • A valid example of an interrupt e.g. 'out of paper' message for a printer	**4**

Question	Answer	Marks
9(a)(i)	**Two** from: • Data is transmitted one bit at a time • Data is transmitted using a single wire • Bits arrive in order/sequence	**2**
9(a)(ii)	**Two** from: • Data is transmitted multiple bits at a time/simultaneously • Data is transmitted using multiple wires • Bits may arrive out of sequence/skewed (and are reordered)	**2**
9(a)(iii)	1 mark for each: • Data is transmitted in both directions • … at the same time/simultaneously	**2**

Question	Answer	Marks
9(b)	Maximum of three marks per error detection method. 1 mark for naming the method, 2 marks for describing it. **Parity (check)** • Odd or even parity can be used • Bits are added together // 1 bits are counted • Parity bit added (depending on parity set) • Parity checked on receipt • If parity bit is incorrect an error is detected **Checksum** • Calculation performed on data (to get the checksum) • Checksum sent with data • Checksum recalculated after transmission • Comparison made between checksum before and checksum after transmission • Error detected if checksums are different **Automatic repeat request (ARQ)** • Uses acknowledgement and timeout • Request is sent (with data) requiring acknowledgement • If no response/acknowledgment within certain time frame data package is resent • When data received contains an error a request is sent (automatically) to resend the data • The resend request is repeatedly sent until packet is received error free/limit is reached/acknowledgement received	**9**

Question	Answer	Marks
10	**Five** from: • The sensor sends data to the microprocessor • The analogue data is **converted to digital** (using ADC) • The microprocessor compares the reading to the set range/stored values/stored data (6 to 8) … – … If the reading is >8 or <6 / outside range … ○ … the **microprocessor** sends a signal to output the alert • The **process** is continuous/repeated	**5**