UNIT 1

MARK SCHEME

Guidance for examiners

Positive marking

It should be remembered that learners are writing under examination conditions and credit should be given for what the learner writes, rather than adopting the approach of penalising him/her for any omissions. It should be possible for a very good response to achieve full marks and a very poor one to achieve zero marks. Marks should not be deducted for a less than perfect answer if it satisfies the criteria of the mark scheme.

For questions that are objective or points-based the mark scheme should be applied precisely. Marks should be awarded as indicated and no further subdivision made.

For band marked questions mark schemes are in two parts.

Part 1 is advice on the indicative content that suggests the range of computer science concepts, theory, issues and arguments which may be included in the learner's answers. These can be used to assess the quality of the learner's response.

Part 2 is an assessment grid advising bands and associated marks that should be given to responses which demonstrate the qualities needed in AO1, AO2 and AO3. Where a response is not credit worthy or not attempted it is indicated on the grid as mark band zero.

Banded mark schemes

Banded mark schemes are divided so that each band has a relevant descriptor. The descriptor for the band provides a description of the performance level for that band. Each band contains marks.

Examiners should first read and annotate a learner's answer to pick out the evidence that is being assessed in that question. Once the annotation is complete, the mark scheme can be applied.

This is done as a two stage process.

Stage 1 – Deciding on the band

When deciding on a band, the answer should be viewed holistically. Beginning at the lowest band, examiners should look at the learner's answer and check whether it matches the descriptor for that band. Examiners should look at the descriptor for that band and see if it matches the qualities shown in the learner's answer. If the descriptor at the lowest band is satisfied, examiners should move up to the next band and repeat this process for each band until the descriptor matches the answer.

If an answer covers different aspects of different bands within the mark scheme, a 'best fit' approach should be adopted to decide on the band and then the learner's response should be used to decide on the mark within the band. For instance if a response is mainly in band 2 but with a limited amount of band 3 content, the answer would be placed in band 2, but the mark awarded would be close to the top of band 2 as a result of the band 3 content. Examiners should not seek to mark candidates down as a result of small omissions in minor areas of an answer.

Stage 2 – Deciding on the mark

Once the band has been decided, examiners can then assign a mark. During standardising (marking conference), detailed advice from the Principal Examiner on the qualities of each mark band will be given. Examiners will then receive examples of answers in each mark band that have been awarded a mark by the Principal Examiner. Examiners should mark the examples and compare their marks with those of the Principal Examiner.

When marking, examiners can use these examples to decide whether a learner's response is of a superior, inferior or comparable standard to the example. Examiners are reminded of the need to revisit the answer as they apply the mark scheme in order to confirm that the band and the mark allocated is appropriate to the response provided.

Indicative content is also provided for banded mark schemes. Indicative content is not exhaustive, and any other valid points must be credited. In order to reach the highest bands of the mark scheme a learner need not cover all of the points mentioned in the indicative content but must meet the requirements of the highest mark band. Where a response is not creditworthy, that is contains nothing of any significance to the mark scheme, or where no response has been provided, no marks should be awarded.

Q	Answer	Mark	AO1	AO2	AO3	Total
1	RAM (any three of) • it has fast read and write access and is volatile • used to store data and currently running programs • RAM is needed because most data on computers is stored in much slower "storage media" such as hard disks, solid state drives or flash memory • For the processor to be able to work on data or run programs at any reasonable speed, the programs or data need to be copied into RAM first. Cache memory (any three of) • is similar to RAM, except it resides on or close to the CPU • is faster than RAM and is also volatile • used to store frequently used data from main memory • used by the processor to avoid having to slow down to the speed of the RAM all the time • usually quite low-capacity (a few megabytes), so RAM is still needed in order to avoid constantly accessing things from slow storage media.	3 3	1.1b 1.1b			6
2	3 marks for functional characteristics of each device x 2 External hard disc drive: • Speed of access – Very fast data transfer, only flash memory is faster • Cost per unit of storage – external hard disc is quite cheap per byte of storage • Portability reason – external hard disc is physically quite small and can be easily stored securely and safely for example in a fire proof safe Third party storage provider: • Speed of access – very fast transfer achievable (depending on network speed) • Cost per unit of storage – could be cheaper or more expensive than external disk – accept either with justification • Data is stored securely and safely on protected servers Flash memory drive: • Speed of access – Very fast transfer which is important for daily updates • Cost per unit of storage – pen drive is quite cheap • Portability reason – pen drive is physically small and can be easily stored securely and safely for example in a fire proof safe Magnetic tape / tape streamer: • Speed of access reason – Access to tape is serial and can be slow but could only back up files amended that day • Cost per unit of storage reason – Tape is relatively	6	1.1b			6

<table>
<tr><th>Q</th><th>Answer</th><th>Mark</th><th>AO1</th><th>AO2</th><th>AO3</th><th>Total</th></tr>
<tr><td></td><td>(but drive can be expensive) cheap compared with other secondary storage mediums
• Portability reason – Tape is physically small and can be easily stored securely and safely for example in a fire proof safe</td><td></td><td></td><td></td><td></td><td></td></tr>
<tr><td>3a</td><td>DHCP - assigning dynamic IP addresses to devices on a network

HTTP - transferring multimedia web pages over the Internet.</td><td>1

1</td><td>1.1b

1.1b</td><td></td><td></td><td>2</td></tr>
<tr><td>3b</td><td>• The file transfer protocol, breaks data into packets and can re-send lost or damaged packets
• it allows packets that have arrived in a random order to be reassembled
• This is convenient for downloading files if network traffic is slows or some of your packets are dropped / arrive out of order
• However, the FTP protocol won't work as well for streaming media as it is more important to continue to receive new packets rather than retransmitting lost or dropped packets

• Voice and video traffic is can be transmitted using UDP
• Real-time video and audio streaming protocols are designed to handle occasional lost packets, so only slight degradation in quality occurs, rather than large delays if lost packets were retransmitted</td><td>1
1
1
1
1
1</td><td></td><td>2.1a
2.1a
2.1a
2.1a
2.1a
2.1a</td><td></td><td>6</td></tr>
<tr><td>3c</td><td>The exchange of signals between devices to establish their readiness to communicate.

Example: Establishing a printers readiness to print</td><td>1

1</td><td>1.1b</td><td>

2.1a</td><td></td><td>2</td></tr>
<tr><td>4a</td><td>AND

<table><tr><th>Input (A)</th><th>Input (B)</th><th>Output (A AND B)</th></tr><tr><td>0</td><td>0</td><td>0</td></tr><tr><td>0</td><td>1</td><td>0</td></tr><tr><td>1</td><td>0</td><td>0</td></tr><tr><td>1</td><td>1</td><td>1</td></tr></table></td><td>1

1</td><td>

1.1a</td><td>2.1a</td><td></td><td>2</td></tr>
<tr><td>4b</td><td>Any one of:
• Picks out / produces right bit / least significant bit (which is 1)
• Determines whether right bit / least significant bit is a 0 or 1</td><td>1</td><td></td><td>2.1a</td><td></td><td>1</td></tr>
<tr><td>5</td><td>Serial transmission: data is sent one bit at a time along the same data line

Advantage (any one of:)
• requires only two wires compared with 8 or 16 in parallel
• serial can travel longer distances than parallel
• simpler interface / circuit board / fewer lines required</td><td>1

1</td><td>1.1a

1.1b</td><td></td><td></td><td>4</td></tr>
</table>

Q	Answer	Mark	AO1	AO2	AO3	Total
	Parallel transmission: all bits in a byte are sent simultaneously along separate lines **Advantage** • transmission is faster than serial transmission	1 1	1.1a 1.1b			
6a	Fragmentation: related data is split and stored on different parts of the disc. If data is fragmented, it takes longer for the disc heads to move between parts of the file, which slows the process of loading it. Defragmentation is the process where files are physically re-arranged on disk so that they are no longer fragmented and the parts of each file are stored together. This improves the speed of accessing data from disk.	1 1 1 1	1.1b 1.1b 1.1b 1.1b			4
6b	Any three of: • SSD uses direct access to data (files) so there would be no improvement in read times as there's no physical read-head to move • Defragmentation may perform "trim" command which may slightly improve the speed of future write operations • SSD is currently made out NAND based flash memory • NAND based flash memory has a limited lifespan – defragmentation process may shorten its lifespan.	3	1.1b			3
7	**Backup** • Backup is a redundant copy of files, usually stored separately from the original system • It can be used to recover data in the event of catastrophic failure of the original storage media **Generations of files** • A generation file backup system involves storage of several of the most recent versions of a master file *Accept grandfather-father-son method* • Useful if one version is corrupted: the previous version(s) is still available **Transaction logs** • A transaction log is used with on-line updating - stores all the update data • It can be used in case of failure - could restore data by being combined with previous master/backup file, with minimal data loss.	 1 1 1 1 1 1	 1.1b 1.1b 1.1b 1.1b 1.1b 1.1b			6
8a	00110110_2 $\underline{00101110_2 +}$ 01100100 Hexadecimal value = 64_{16}	1 1		2.1a 2.1a		2
8b	10001100_2	1		2.1a		1
8c	11110101_2 One method is: • From RHS, rewrite it up to and including the first one	1 1	 1.1b	2.1a		3

Q	Answer	Mark	AO1	AO2	AO3	Total
	• Change other 1 digits to 0 and 0 digits to 1 Alternatively • Flip the bits • Add one • (Ignore carry (ninth bit)) (Other methods equally acceptable)	1	1.1b			
8di	Any one of: • are not normally stored accurately • require more complex processing • no exact representation of zero	1	1.1b			1
8dii	000101111100_2 0101_2 1 for correct mantissa, 1 for correct exponent	2		2.1a		2
8diii	• Mantissa = 0.6875_{10} or 11/16, Exponent = 5_{10} • Answer = Mantissa x $2^{exponent}$ • Answer = 22_{10}	1 1 1		2.1a 2.1a 2.1a		3
9a	A.(B + C) (A.B) + (A.C)	 1		 2.1a		1
9b	A.B + A.(B + C) + B.(B + C) A.B + A.B + A.C + B.B + B.C A.B + A.B + A.C + B + B.C A.B + A.C + B + B.C A.B + A.C + B B + A.C	 1 1 1 1 1		 2.1a 2.1a 2.1a 2.1a 2.1a		5
10a	OR	1		2.1a		1

10b	**Example**
	`24 if Num MOD Divisor = 0 then`
	`25  set Prime = FALSE`
	`26 endif`
	
	OR
	
	`17 if Prime = TRUE then`
	`18  output Num, "is a prime number"`
	`19  else`
	`20  output Num, "is NOT a prime number"`
	`21  Endif`
	
	The purpose of selection is to execute code if a certain condition is met.
		1
		
		1
			
			1.1b
				2.1a		2

10c	**Example**
	`10 repeat`
	`11  if Num MOD Divisor = 0 then`
	`12  set Prime = FALSE`
	`13  endif`
	`14  set Divisor = Divisor + 1`
	`15 until (Prime = FALSE) OR (Divisor = Num)`
	
	The purpose of repetition is to repeatedly execute code until a certain condition is met.
		1
		
		1
			
			1.1b
				2.1a		2

10d	• correct prime numbers and output " "x is a prime number" e.g. "3 is a prime number" • correct non-prime number and output "x is NOT a prime number" , e.g. " 4 is not a prime number"	1 1			3.1c 3.1c	2
11a	Ask customer to type password twice and compare both	1		2.1a		1

<table>
<tr><th>Q</th><th>Answer</th><th>Mark</th><th>AO1</th><th>AO2</th><th>AO3</th><th>Total</th></tr>
<tr><td></td><td>inputs to check that they are the same</td><td></td><td></td><td></td><td></td><td></td></tr>
<tr><td>11b</td><td>1 mark for check correctly described
1 mark for each example of invalid data that the check described would detect

<table>
<tr><th>Suitable checks</th><th>Example of invalid data</th></tr>
<tr><td>Range check – entries between sensible limits, e.g. 0-60</td><td>-1 or 74</td></tr>
<tr><td>Type check – all entries should be integer</td><td>B or #</td></tr>
</table>

NOTE - Do not accept length check, format check or look up check and examples of invalid data must follow check described</td><td>1
1</td><td></td><td>2.1a
2.1a</td><td></td><td>2</td></tr>
<tr><td>11c</td><td>1 mark for check correctly described
1 mark for each example of invalid data that the check described would detect

<table>
<tr><th>Suitable checks</th><th>Example of invalid data</th></tr>
<tr><td>Format check - email address has a string@string.string</td><td>abczyz$email.co.uk</td></tr>
</table>

NOTE - Do not accept length check, type check or look up check and examples of invalid data must follow check described</td><td>1
1</td><td></td><td>2.1a
2.1a</td><td></td><td>2</td></tr>
<tr><td>12</td><td>

```
declare Rainfall array (1..12) of
integer
set Total = 0

for Count = 1 to 12
   input Rainfall(Count)
   set Total = Total + Rainfall(Count)
endfor

set Mean = Total / 12

output "Total = ", Total
ouput "Mean = ", Mean

output "Months above Mean = "
for Count = 1 to 12
   if Rainfall(Count) > Mean then
         output Count
Endfor
```

Marking

Declare array and initialise variables
Input loop structure + increment
Calculate mean
Output Total and Mean
Output loop structures
Detect and output above mean months
</td><td>1
1
1
1
1
1</td><td></td><td></td><td>3.1b
3.1b
3.1b
3.1b
3.1b
3.1b</td><td>6</td></tr>
<tr><td>13a</td><td>• Alpha testing – when software is issued to a restricted</td><td>1</td><td>1.1b</td><td></td><td></td><td>3</td></tr>
</table>

Q	Answer	Mark	AO1	AO2	AO3	Total
	audience of testers within the developer's own company • Beta testing - when a version is released to a number of people external to the company e.g. privileged customers in exchange for their constructive comments • Acceptance testing - when testing is carried out to prove to the customer / end user that the system works correctly.	1 1	1.1b 1.1b			
13b	• Perfective - is when the performance/functionality of the program has to be enhanced • Adaptive – is when the program has to be altered e.g. to run on a different operating system • Corrective – is while the program is being used and an error is discovered and corrected	1 1 1	1.1b 1.1b 1.1b			3
14	For each stage, 1 mark for each bullet point up to a maximum of 2 marks No marks for simply naming stages Lexical analysis • Comments and unneeded spaces are removed • Keywords, constants and identifiers are replaced by 'tokens' • A symbol table is created which holds the addresses of variables, labels and subroutines Syntax analysis • Tokens are checked to see if they match the spelling and grammar expected, using standard language definitions. This is done by parsing each token to determine if it uses the correct syntax for the programming language. • If syntax errors are found, error messages are produced Semantic analysis • Variables are checked to ensure that they have been properly declared and used • Variables are checked to ensure they are of the correct data type, e.g. real values are not being assigned to integers • Operations are checked to ensure that they are legal for the type of variable being used e.g. you would not try to store the result of a division operation as an integer Code generation • Machine code is generated • Code optimisation may be employed to make it more efficient / faster / less resource intense	8	1.1b			8
15	**Indicative content** • Data compression reduces the file size Lossy data compression • Compressed files can never be recovered exactly as	8	1.1b			8

Q	Answer	Mark	AO1	AO2	AO3	Total
	they were before they were compressed • When compressed files are decompressed they do not give back the original data, i.e. data is lost • Because lossy compression cannot be decompressed to yield the exact original data, it is not a good method of compression for critical data, such as textual data • It is most useful for digitally sampled analogue data, such as sound, video, graphics or images • Algorithms for lossy compression vary, but many use a threshold level truncation. This means that a level is chosen past which all data is truncated, e.g. in a sound file, the very high and low frequencies, which the human ear can not hear, may be truncated from the file • Some examples of lossy data compression algorithms are JPEG, MPEG, and MP3. Lossless data compression • The original message can be decompressed back to its original form (recovers all original data) • Lossless data compression works by finding repeated patterns in data and compressing those patterns in an efficient manner. For this reason, lossless data compression is also referred to as redundancy reduction. Becuase redundancy reduction is dependent on patterns in the message, it does not work well on random messages. Lossless data compression is ideal for text. Most of the algorithms for lossless compression are based on the LZ compression method developed by Lempel and Ziv. • One type of text encoding which is very effective for files with long strings of repeating bits is RLE. RLE stands for Run Length Encoding • RLE uses a sliding dictionary method of the LZ algorithm. The sliding dictionary method utilizes pointers within the compressed file that point to previously represented strings of bits within the file. • Here is an example of a message which could be effectively encoded with RLE: o The word the, is the most frequently used word in the English language. The string "the" could be represented only once and could be pointed to by all later calls to that string • Huffman coding works by analyzing the frequency of elements in data. The elements with the highest frequency get assigned the shortest encoding (with the fewest bits). Elements with lower frequencies get assigned longer encodings (with more bits) • Huffman coding could be used to compress sound files, particulary recordings containing frequecies of that heard in a human voice. *Other compression techniques accepted.*					

Band	AO1.1b Max 8 marks
3	**7 - 8 marks** The candidate has: • written an extended response that has a sustained line of reasoning which is coherent, relevant, and logically structured • shown clear understanding of the requirements of the question and a clear knowledge of the indicative content. Clear knowledge is defined as a response that provides seven to eight relevant detailed points on lossy and lossless data compression techniques, which relate to an extensive amount of the indicative content • addressed the question appropriately with minimal repetition and no irrelevant material • has presented a balanced discussion and justified their answer with examples • used appropriate technical terminology referring to the indicative content confidently and accurately.
2	**3 - 6 marks** The candidate has: • written a response that has an adequate line of reasoning with elements of coherence, relevance, and logical structure • shown adequate understanding of the requirements of the question and a satisfactory knowledge of the topic of changeover as specified in the indicative content. Satisfactory knowledge is defined as a response that provides three to six points on lossy and lossless data compression techniques as signalled in the indicative content. Up to five marks could be awarded to a response that provides detailed points on one data compression techniques (lossy or lossless) • has presented a discussion with limited examples • used appropriate technical terminology referring to the indicative content.
1	**1 - 2 marks** The candidate has: • written a response that that lacks sufficient reasoning and structure • produced a discussion which is not well developed • attempted to address the question but has demonstrated superficial knowledge of the topics specified in the indicative content. Superficial knowledge is defined as a response that provides one to two points on lossy and lossless data compression techniques as signalled in the indicative content • used limited technical terminology referring to the indicative content.
0	**0 marks** Response not credit worthy or not attempted.