Cambridge International AS & A Level

Cambridge Assessment International Education
Cambridge International Advanced Subsidiary and Advanced Level

COMPUTER SCIENCE **9608/13**

Paper 1 Written Paper **May/June 2019**

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 May/June 2019 series for most Cambridge IGCSE™, Cambridge International A and AS Level and Cambridge Pre-U components, and some Cambridge O Level components.

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.

<table>
<tr><th>Question</th><th>Answer</th><th>Marks</th></tr>
<tr><td>1(a)</td><td>1 mark for each correct entry

<table>
<tr><th>Management task</th><th>Description</th></tr>
<tr><td>Memory management</td><td>Handles the allocation of memory to processes // Ensures two programs do not attempt to use the same memory locations // Keeps track of allocated and free memory locations</td></tr>
<tr><td>Security management</td><td>Provides user accounts and passwords</td></tr>
<tr><td>Interrupt processing</td><td>Handles the signals sent when the attention of the processor is required elsewhere</td></tr>
<tr><td>Provision of a software platform</td><td>Provides an environment within which programs can be run</td></tr>
</table></td><td>4</td></tr>
<tr><td>1(b)(i)</td><td>1 mark per bullet point to max 2 for formatter, max 2 for defragmenter

hard disk formatter
Makes existing data inaccessible
Partitions the disk into logical drives
Sets up the (specified) file system
Prepares the disk for initial use
May check for errors on the disk

hard disk defragmenter
Re-organises the disk contents
Moves split files so they are contiguous
Creates a larger area of (contiguous) free space</td><td>4</td></tr>
<tr><td>1(b)(ii)</td><td>1 mark per bullet point

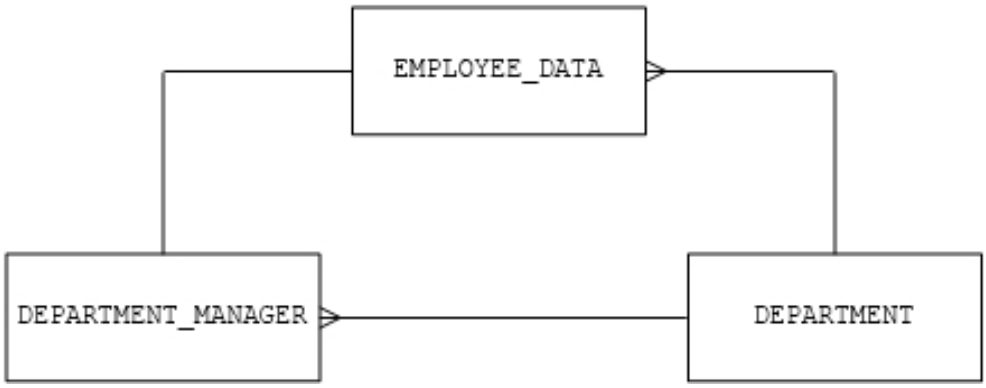
For example:

Backup software
File compression
Virus checker
Disk contents analysis / repair</td><td>3</td></tr>
</table>

Question	Answer	Marks
2(a)	**1 mark** per bullet point Security: keep data safe from accidental/malicious damage/loss Privacy: keep data confidential // only seen by authorised personnel	**2**
2(b)	**1 mark** for identifying method **2 marks** for description to **max 2 × 3** For example: User accounts User has a username and password Access to resources can be limited to specific accounts Cannot access system without valid username and password // prevents unauthorised access to the system Firewall All incoming and outgoing network traffic goes through firewall Blocks signals that do not meet requirements Keeps a log of signals Applications can have network access restricted Anti-malware Scans for malicious software Quarantines or deletes any malicious software found Scans can be scheduled at regular intervals Should be kept up to date Auditing Logging all actions/changes to the system In order to identify any unauthorised use Application Security (accept equivalent) Applying regular updates / patches Finding, fixing and preventing security vulnerabilities in any (installed) application	**6**
2(c)(i)	**1 mark** per bullet to **max 4** Each byte has a parity bit // horizontal parity An additional parity byte is sent with vertical (and horizontal) parity Each row and column must have an even/odd number of 1s Identify the incorrect row and column The intersection is the error	**4**
2(c)(ii)	**1 mark** for correct answer Errors in an even number of bits (could cancel each other out)	**1**

Question	Answer	Marks
2(d)	**1 mark** per bullet point to **max 3** For example: He should ... ...Keep the client's personal data private ...Involve the client in the development // …Communicate with the client …Provide the solution that the client asked for ...Keep the project running on time // budget ...Keep the client informed of any problems/delays	**3**
2(e)	**1 mark** per bullet point to **max 3** He uses a compiler because it creates a separate executable file The executable means the client cannot access the source code // edit the program The executable means the client does not need the IDE / compiler It will (probably) be faster to run the executable than to interpret the source code every time the program is run	**3**

Question	Answer	Marks
3(a)(i)	**1 mark** per bullet point Stores all the information about the database // data about the data // metadata about the data For example, fields, data types, validation, keys	**2**
3(a)(ii)	**1 mark** per bullet point to **max 2** Allows the user to enter criteria Searches for data which meets the entered criteria Organises the results to be displayed to the user	**2**
3(b)	**1 mark** per bullet point to **max 2** Primary key uniquely identifies each tuple // Each tuple in the table is unique Primary key can be used as a foreign key in another table ….. to form a link/relationship between the tables By example: Identification of a primary key in a table Describing <u>that</u> primary key in another table as a foreign key	**2**

Question	Answer	Marks
3(c)	**1 mark** for each correct link  	**3**
3(d)	**1 mark** per bullet point to **max 3** There are no repeating groups (1NF) There are no partial dependencies (2NF) There are no non-key dependencies // There are no transitive dependencies (3NF)	**3**
3(e)(i)	**1 mark** for correct answer `CREATE DATABASE EMPLOYEES;`	**1**

3(e)(ii)	**1 mark** per bullet
	
	Create table `EMPLOYEE_DATA` with open and close brackets
	`EmployeeID` as `VarChar` restricted to max 7, `Gender` as a `VarChar` restricted to max 6, `DepartmentNumber` as a `VarChar` restricted to max 2
	`FirstName` and `LastName` as `VarChar` (any max lengths must be reasonable)
	Date of birth as `Date`
	Declaring `EmployeeID` as PK
	
	`CREATE TABLE EMPLOYEE_DATA (`
	`  EmployeeID VarChar(7),`
	`  FirstName VarChar,`
	`  LastName VarChar,`
	`  DateOfBirth Date,`
	`  Gender VarChar(6),`
	`  DepartmentNumber VarChar(2),`
	`  PRIMARY KEY (EmployeeID)`
	`);`
		5

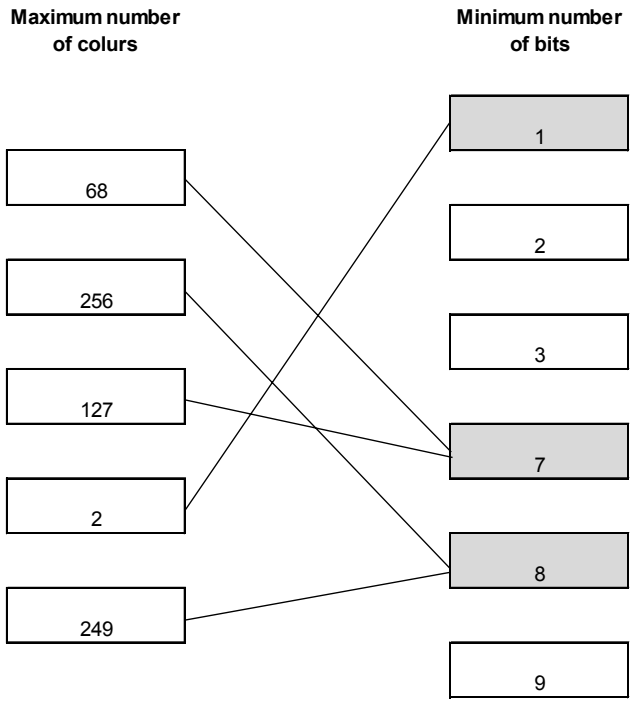
Question	Answer	Marks

3(e)(iii)	**1 mark** per bullet
	
	Select FirstName and LastName only
	From both tables
	Where DepartmentName = “Finance”
	AND Gender = “Female”
	Joining tables (either AND, or inner join)
	
	`SELECT FirstName, LastName`
	`FROM EMPLOYEE_DATA, DEPARTMENT`
	`WHERE DepartmentName = "Finance"`
	`AND Gender = "Female"`
	`AND DEPARTMENT.DepartmentNumber =`
	`EMPLOYEE_DATA.DepartmentNumber;`
		5

Question	Answer	Marks
4(a)	**1 mark** per bullet LDM #300 The (denary) number 300 is loaded (into the register) LDD 300 The contents of address 300 are loaded (into the register)	**2**
4(b)	**1 mark** for JPN correct **1 mark** for both of ADD and DEC correct **1** mark for LDR correct	**3**

	Description	Jump instruction	Arithmetic operation	Data movement
`LDR #3`	Load the number 3 to the Index Register			✓
`ADD #2`	Add 2 to the Accumulator		✓	
`JPN 22`	Move to the instruction at address 22	✓		
`DEC ACC`	Subtract 1 from the Accumulator		✓	

Question	Answer	Marks
4(c)(i)	**1 mark** for hardware interrupt **1 mark** for software interrupt For example: Hardware interrupt Printer out of paper No CD in drive Software interrupt A running program needs input Runtime error, e.g. division by zero	**2**
4(c)(ii)	**1 mark** per bullet to **max 5** At the start / end of each fetch-execute cycle the processor checks for interrupt(s) Check if an interrupt flag is set // Check if bit set in interrupt register Processor identifies source of interrupt Processor checks priority of interrupt If interrupt priority is high enough // Lower priority interrupts are disabled Processor saves current contents of registers // saves current job on stack Processor calls interrupt handler / Interrupt Service Routine (ISR) Address of ISR is loaded into Program Counter (PC) When servicing of interrupt complete, processor restores registers // job from stack is restored Lower priority interrupts are re-enabled Processor continues with next F–E cycle	**5**

Question	Answer	Marks
5(a)	**1 mark** for each correctly linked box on the right  	**3**
5(b)(i)	**1 mark** for correct answer 40 images or frames are displayed/recorded each second	**1**
5(b)(ii)	**1 mark** per bullet to **max 4** Progressive – each frame contains (all the lines for) the complete image Interlaced – each frame contains half the (number of lines) of the complete image Progressive – frames are not divided into fields Interlaced – each frame divided into two fields // One field contains the data for the even numbered rows / lines and the other has the data for the odd numbered rows / lines. Progressive – complete frames are displayed in sequence Interlaced – data from two frames is displayed simultaneously Progressive – the number of images stored is the same as the frame rate // the rate of picture display is the same as the frame rate. Interlaced – the rate of picture display (the field rate) is twice the rate of image frame display (the frame rate). Progressive means the entire frame is refreshed each time Interlaced means only half the frame is refreshed Progressive has high bandwidth requirements Interlaced halves the transmission bandwidth requirements.	**4**

Question	Answer	Marks
5(c)	**1 mark** per bullet 88.2 kHz The sound wave is sampled 88200 times per second 32 bits Each sample is stored as a 32-bit binary number	**2**