National
Qualifications
RESOURCE

X816/76/01 Computing Science

Marking Instructions

Please note that these marking instructions have not been standardised based on candidate responses. You may therefore need to agree within your centre how to consistently mark an item if a candidate response is not covered by the marking instructions.

General marking principles for Higher Computing Science

This information is provided to help you understand the general principles you must apply when marking candidate responses to questions in this paper. These principles must be read in conjunction with the detailed marking instructions, which identify the key features required in candidate responses.

(a) Marks for each candidate response must **always** be assigned in line with these general marking principles and the detailed marking instructions for this assessment.

(b) Always use positive marking. This means candidates accumulate marks for the demonstration of relevant skills, knowledge and understanding; marks are not deducted.

(c) If a candidate response is not covered by either the principles or detailed marking instructions, and you are uncertain how to assess it, you must seek guidance from your team leader.

(d) Award marks regardless of spelling, as long as the meaning is unambiguous. This applies to all responses, including code. Award marks as per the detailed marking instructions, regardless of syntax errors, if the intention of the coding is clear.

(e) For questions where candidates are asked to design or write code, a sample response is shown in the detailed marking instructions. This will not be the only valid response. You must use the detailed marking instructions and additional guidance to ensure that you consider alternative approaches and nuances of different programming languages. If in doubt you should refer to your Team Leader.

(f) A correct response can be negated if the candidate includes an extra, incorrect response. For example, if the candidate is asked for two answers for two marks and the candidate gives three, one of which is incorrect, they are awarded one mark.

(g) If a candidate scores through a response and makes a further attempt, you should only mark the further attempt. If no further attempt is made and the original is legible, you should mark the original response.

(h) Where an incorrect response is carried forward and used correctly in a following part of the question, you should give credit for subsequent responses that are correct with regard to the original error. Candidates should not be penalised more than once for the same error.

(i) Only award marks for a valid response to the question asked. Where candidates are asked to:

- **Identify**, **name**, **give** or **state,** they need only name or present in brief form.
- **describe**, they must provide a statement or structure of characteristics and/or features. This will be more than an outline or a list. It may refer to, for example, a concept, process, experiment, situation, or facts, in the context of and appropriate to the question. Candidates must make the same number of factual/appropriate points as there are marks available in the question.
- **explain**, they must relate cause and/or effect and/or make relationships between things clear, in the context of the question or a specific area within the question.
- **write code,** they must write recognisable code, not prose nor a diagram.
- **design,** they must use a design technique appropriate to the problem. Award marks as per the detailed marking instructions, regardless of errors in the exemplification of the technique, if the intention of the design is clear.

(j) In the detailed marking instructions, if a word is underlined then it is essential; if a word is bracketed() then it is not essential. Words separated by / are alternatives

Marking instructions for each question

Section 1 — Software design and development and computer systems

Question			Expected response	Max mark	Additional guidance
1.	(a)		1111 0111	**1**	
	(b)		-128 to 127 **OR** -2^7 to 2^7-1	**2**	
2.	(a)		• increase the number of cores • increase width of data bus • add cache/increase cache • any other valid response	**1**	**1 mark** for any one bullet
	(b)		Matching explanation • multiple instructions simultaneously • more bits transferred in a single operation • reduces number of accesses to slower main memory • any other valid response	**1**	**1 mark** for bullet that matches method stated in part (a) Do not award the mark for increasing clock speed as it is in the stem.
3.			• autonomous driving (1) is more fuel efficient due to system controlling accelerating/ decelerating and detecting/anticipating braking (1) • intelligent route planning (1) reduces driving time by monitoring external factors such as accidents/volume of traffic which reduces fuel consumption (1) • tracking parking (1) reduces driving time searching for space and therefore fuel consumption (1) • engine management system (1) optimises engine efficiency reducing fuel consumption (1) • intelligent road traffic management systems (1) adjusting speed limits to optimise traffic flow reducing fuel consumption (1) • any other valid response	**2**	**1 mark** for feature **1 mark** for justification related to fuel

Question			Expected response	Max mark	Additional guidance
4.			• Sign bit: 1 • Remaining 15 bits of mantissa: 110 0000 0000 0000 • Exponent: 1111 1110	**3**	**1 mark** for each bullet
5.			• authenticates the sender • guarantees the integrity of the sent item/sent item has not been altered	**2**	**1 mark** for each bullet OR Award second mark if candidate describes mechanism for integrity or authenticity. For example • generate a code/hash for the document and when received the code/hash is recalculated and compared OR • code/hash is encrypted with sender's private key and decrypted using the public key

6.	(a)		• Condition for length of string
			• Condition for count of valid characters
				2	**1 mark** for each bullet
					
					Example answers
					
					`IF lengthCheck =true AND countValid=6 THEN`
					
					`IF length(dob)=6 and countValid=6 THEN`

	(b)		• checks characters even if the length <>6 • keeps checking characters even when an invalid character is found	**1**	**1 mark** for any one bullet Do not accept generic answers relating to efficiency. Answer must be in context and refer to the code.

Question			Expected response	Max mark	Additional guidance

7.	(a)	(i)	• record structure with name
			• four fields specified
				2	**1 mark** for each bullet
					
					Record structure could be Type, Structure, Class, Named Tuple, Record name IS
					
					Example Answers:
					
					SQARL
					`RECORD competitorData IS ( STRING name, STRING club, STRING event, REAL distance)`
					
					VB
					`Private/Public/Structure Type competitorData`
					`  name as string`
					`  club as string`
					`  event as string`
					`  distance as single`
					`END TYPE`
					
					Livecode
					`Global CompetitorDATA`
					`  Put empty into CompetitorData[1][name]`
					`Put empty into CompetitorData[1][club]`
					`Put empty into CompetitorData[1][event]`
					`Put empty into CompetitorData[1][distance]`
					
					Python Class method
					`class Competitor ():`
					`  def __init__(self):`
					`  self.name = ""`
					`  self.club = ""`
					`  self.event = ""`
					`  self.distance = 0.0`
					
					Python Dataclass method
					`Class Competitor:`
					`  name: str = ""`
					`  club: str = ""`
					`  event: str = ""`
					`  distance: float = 0.0`

Question			Expected response	Max mark	Additional guidance

7.	(a)	(ii)	• array structure
			• using data type created in (i)
				2	**1 mark** for each bullet
					
					Example Answers:
					
					Python
					`competitors=[datatype_a() for x in range (800)]`
					
					VB
					`DIM competitors(799) AS datatype_a`
					
					Livecode
					`Set the data of EventData to CompetitorData`

	(b)		• open and close file
			• loop traversing the array with termination
			• IF with condition event="Javelin"
			• AND condition distance>=70
			• writing name and club to file
				5	**1 mark** for each bullet
					
					Open file
					Loop for all 800/to end of array
					If event="Javelin" AND distance>=70 then
					Write name, club to file
					End if
					End loop
					Close file

Question			Expected response	Max mark	Additional guidance
7.	(c)		• ask for searchName/searchEvent and their use in the IF statement • loop that traverses correctly with termination IF condition with: • correct use of array variable • comparison of current name and event fields to targets • display if found with correctly concatenated message • display if not found using any suitable flag, for example -1 or found is false	**6**	**1 mark** for each bullet Candidates' solutions will vary in terms of efficiency but marks are awarded for any correct implementation.

Example answers:

```
RECEIVE searchName FROM KEYBOARD
RECEIVE searchEVENT FROM KEYBOARD
DECLARE found INITIALLY false
SET counter TO -1
REPEAT
  counter=counter+1
  IF searchName=competitors[counter].name AND
searchEvent=searchName=competitors[counter].event then
     SET found TO true
  END IF
UNTIL found=true OR counter=799
IF found =true THEN
  SEND competitors[counter].name & " threw "
&competitors[counter].distance & " in the " &
competitors[counter].event TO DISPLAY
ELSE
  SEND " Competitor not found" TO DISPLAY
END IF
```

Using a fixed loop:

```
RECEIVE searchName FROM KEYBOARD
RECEIVE searchEVENT FROM KEYBOARD
DECLARE found INITIALLY false
FOR index FROM 0 TO 799 DO
  IF searchName=competitors[index].name AND
searchEvent=competitors[index].event then
    SEND competitors[index].name & " threw " &
competitors[index].distance & " in the " &
competitors[index].event TO DISPLAY
    SET found TO true
  END IF
END FOR
IF found =false THEN
     SEND 'Competitor not found' TO DISPLAY
END IF
```

<table>
<tr><th colspan="3">Question</th><th>Expected response</th><th>Max mark</th><th>Additional guidance</th></tr>
<tr><td>8.</td><td>(a)</td><td></td><td>STEP 2 IN: password
STEP 3 IN: randPos1, randPos2, randPos3</td><td>2</td><td>1 mark for each step</td></tr>
<tr><td></td><td>(b)</td><td></td><td>Returns a value</td><td>1</td><td></td></tr>
<tr><td></td><td>(c)</td><td></td><td>The data flow informs parameter passing.</td><td>1</td><td>The mark is for identifying the relationship to parameter passing. Candidates that use actual or formal incorrectly here should not be penalised as part (d) assesses the use of formal and actual parameters.</td></tr>
<tr><td></td><td>(d)</td><td>(i)</td><td>• correct identification of any formal parameter
• corresponding actual parameter</td><td>2</td><td>1 mark for each bullet
<table><tr><th>Formal</th><th>Actual</th></tr><tr><td>word</td><td>password</td></tr><tr><td>first</td><td>ranPos1</td></tr><tr><td>second</td><td>ranPos2</td></tr><tr><td>third</td><td>ranPos3</td></tr></table></td></tr>
<tr><td></td><td></td><td>(ii)</td><td>Local / only in the getLetters sub-program</td><td>1</td><td>Award mark if candidate references appropriate line numbers.</td></tr>
<tr><td></td><td></td><td>(iii)</td><td>• use of conditional loop
• use of comparison to number(s) already generated</td><td>2</td><td>1 mark for each bullet

Candidates may choose to exemplify through the use of design or code, for example.

Do
SET second TO <random number…>
Loop until second<>first
Do
Set third TO <random number…>
Loop until third<>second AND third<>first</td></tr>
<tr><td></td><td>(e)</td><td></td><td>• allows fewer lines of code/removes repetition (1) by calling same procedure more than once (with different actual parameters) (1)
• local variables can use less memory (1) as they only allocated memory during the running of the procedure / memory space is freed at the end of the procedure (1)</td><td>2</td><td>1 mark for reason which relates to procedures.

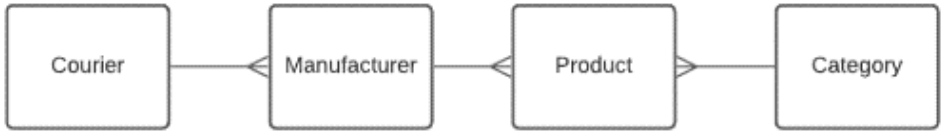
1 mark for justification.

Do not penalise candidates that refer to functions.</td></tr>
</table>

Question			Expected response	Max mark	Additional guidance
9.	(a)		• count number of matching restaurants • find highest rated restaurant	**2**	**1 mark** for each bullet
	(b)		• initialising total and incrementing total by 1 • loop that traverses array with termination IF statement with • condition for food type AND • condition for city	**4**	**1 mark** for each bullet Pseudocode example: (Get targetFood from user Get targetCity from user) Set count to 0 Start loop for each restaurant If restaurantType[index] = targetFood AND city[index] = targetCity Set count to count +1 End if End loop
	(c)	(i)	• A=4.12 • B=4.99 • C=false	**3**	**1 mark** for each bullet
		(ii)	• a breakpoint will halt execution of the code at a predefined point (various line numbers are suitable) • then the values of variable can be inspected to compare with trace table values/expected values	**2**	**1 mark** for each bullet Award second mark if candidates refer to stepping through the code to compare with trace table values as this is another expression of second bullet point.
		(iii)	Change > to < **OR** swap highest and list[index]	**1**	
	(d)		• function called correctly with parameter • assigning result to myHighest	**2**	**1 mark** for each bullet `SET myHighest TO findMax(rating)`

Section 2 — Database design and development

<table>
<tr><th colspan="3">Question</th><th>Expected response</th><th>Max mark</th><th>Additional guidance</th></tr>
<tr><td>10.</td><td></td><td></td><td>See entity-relationship diagram below.</td><td>3</td><td>1 mark for each relationship

• Courier – Manufacturer (1:M)
• Manufacturer – Product (1:M)
• Product – Category (M:1)</td></tr>
<tr><td></td><td></td><td></td><td colspan="3"> 
</td></tr>
<tr><td>11.</td><td></td><td></td><td>• grouping on rating
• corresponding values for each rating
• sort in descending order</td><td>3</td><td>1 mark for each bullet

<table><tr><th>rating</th><th>Longest movie</th></tr><tr><td>PG</td><td>143</td></tr><tr><td>12A</td><td>130</td></tr><tr><td>15</td><td>119</td></tr></table></td></tr>
<tr><td>12.</td><td>(a)</td><td></td><td>Field(s) and Calculation(s)
• Use of both type and alias [Most Expensive Item]
• Maximum aggregate function on the price field

Search Criteria
• Use of complex condition / wildcard

Grouping
• type</td><td>4</td><td>1 mark for each bullet

Search criteria can also use a wildcard, for example type LIKE “*bag”

Allow use of % for wildcard.

Allow use of ‘maximum’ in the design of a query.</td></tr>
<tr><td></td><td></td><td></td><td colspan="3">Example answer:

<table><tr><td>Field(s) and calculation(s)</td><td>type, MAX(price) AS [Most Expensive Item]</td></tr><tr><td>Tables(s) and query</td><td>Snack</td></tr><tr><td>Search criteria</td><td>WHERE type = "Multi-bag" or type="Family bag"</td></tr><tr><td>Grouping</td><td>type</td></tr><tr><td>Sort order</td><td></td></tr></table></td></tr>
</table>

Question			Expected response	Max mark	Additional guidance

12.	(b)		• UPDATE Snack table
			• SET price to new value (price*0.5 OR price/2)
			• WHERE criteria based on stock level
				3	**1 mark** for each bullet
					
					Example answer:
					
					`UPDATE Snack`
					`SET price = 0.5*price`
					`WHERE stock > 500`

	(c)		• SELECT clause with type and SUM(stock)
			• Alias as [Total Stock] with FROM and GROUP BY unchanged
			• ORDER BY with sum(stock) in descending order
				3	**1 mark** for each bullet
					
					Example answer:
					
					`SELECT type, sum(stock) AS [Total Stock]`
					`FROM Snack`
					`GROUP BY type`
					`ORDER BY sum(stock) DESC`

	(d)		• DELETE FROM Snack
			• WHERE criteria on product name uses both wildcards
				2	**1 mark** for each bullet
					
					Example answers:
					
					`DELETE FROM Snack`
					`WHERE productName LIKE "*Salt*";`
					
					`Delete FROM Snack`
					`WHERE productName LIKE "%Salt%";`
					
					Note `"*salt*" or "%salt%"` is allowed.

<table>
<tr><th colspan="3">Question</th><th>Expected response</th><th>Max mark</th><th>Additional guidance</th></tr>
<tr><td>13.</td><td>(a)</td><td></td><td>A customer may rent the same item more than once (which would not be unique).</td><td>1</td><td></td></tr>
<tr><td></td><td>(b)</td><td></td><td>Field(s) and Calculation(s)
• Use of three fields customerID, itemType and startDate, and the calculated field (noOfDays * dailyRentalPrice)

Tables
• Two tables i.e. Rental, Items

Search Criteria
• Use of complex condition OR of wildcard for July 2020 dates

Sort order
• startDate with correct sort order

Example answer
<table><tr><td>Field(s) and calculation(s)</td><td>customerID, itemType, startDate, (noOfDays * dailyRentalPrice)</td></tr><tr><td>Tables(s) and query</td><td>Rental, Item</td></tr><tr><td>Search criteria</td><td>AND startDate LIKE "*/07/2020"
OR
AND startDate LIKE "%/07/2020"

(AND Item.itemID = Rental.itemID
AND Customer.customerID = Rental.customerID)</td></tr><tr><td>Grouping</td><td></td></tr><tr><td>Sort order</td><td>startDate ASC</td></tr></table></td><td>4</td><td>1 mark for each bullet

Equi-join not needed in design.

Search criteria could be expressed as startDate>=“01/07/2020” AND startDate<=“31/07/2020”

Allow use of # around date.

As SQL defaults to sorting by ascending order, ASCs does not need to be specified.</td></tr>
<tr><td></td><td>(c)</td><td></td><td>• FROM clause includes Item and MaxRental
• WHERE clause is dailyRentalPrice = [Max daily rental]</td><td>2</td><td>Example answer

<pre>SELECT itemType, itemModel
FROM Item, MaxRental
WHERE dailyRentalPrice =
[Max daily rental];</pre></td></tr>
</table>

Section 3 - Web design and development

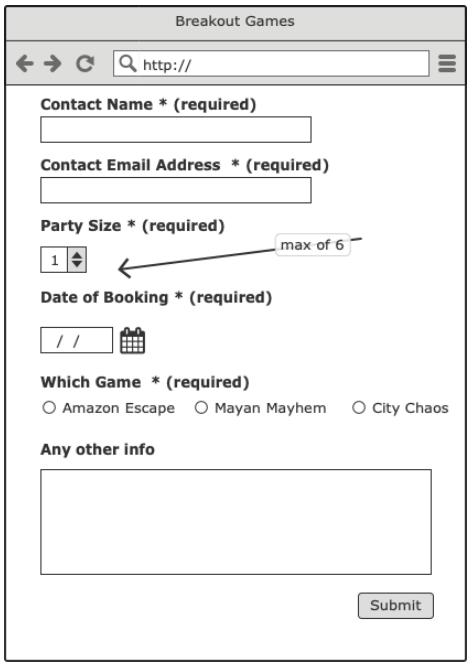
Question			Expected response	Max mark	Additional guidance
14.	(a)		• Appropriate function name with parameter in brackets and subsequent use of parameter for colour change to black	**1**	Example answer `function displayBlack(my_text) { my_text.style.color="Black" ; }`
	(b)		• Add onmouseout event to element • Match function name to name created in part (a)	**2**	**1 mark** for each bullet Example answer `<p onmouseover="displayRed(this)" onmouseout="displayBlack (this)" > Sale now on</p>`

15.			• Grouping of floorplan and event to width 700px
			• Grouping gaminglogo and eventlogo to width/height 75 px, float right and margin-right 20 px
			• Implementing the 10 px margins in any of the grouping selectors
				3	**1 mark** for each bullet
					
					Example answers:
					
					`Header,nav,footer { margin-left 10px;margin-right: 10px;} #gaminglogo,eventlogo { width: 75px; height: 75px; float:right; margin-right: 20px;} #event,#floorplans {width: 700px; margin-left: 10px; }`
					
					OR
					
					`Header, nav, footer{margin-right:10px} Header, #event, footer, nav, #floorplan{margin-left:10px} #gaminglogo, #eventlogo{width:75px; height:75px; float:right; margin-right:20px} #floorplan, #event{Width:700px}`

Question			Expected response	Max mark	Additional guidance
16.	(a)	(i)	Submit (button)	**1**	
		(ii)	`<input type="submit">`	**1**	Additional attributes for the input are not required.
	(b)	(i)	Checked	**1**	
		(ii)	`<select name="instrument" size="3" multiple>`	**1**	
	(c)		• persona may be that of a non-technical, reading ability, accessibility issue • allows developer to assess/observe/gain feedback on aspects of the website for example navigation, use of form, visual layout etc.	**2**	Do not accept young, elderly or age on its own.
	(d)		• the heading is a block element • which takes up all of the width of its container element • changing it to an inline element allows the heading and the image to take up the width it requires	**2**	**1 mark** for each bullet Maximum 2 marks

Question			Expected response	Max mark	Additional guidance

17.	(a)		`.eventimage { margin-top: 10px; margin-left: 10px ; margin-right: 20px ;}`
			OR
			`.eventimage { padding-top: 10px; padding -left: 10px ; padding -right: 20px ;}`
				2	All 3 properties for 2 marks
					any 2 properties for 1 mark

	(b)		• submit button And • text fields with validation indicated e,g, required or use of * for contact name, email and also with text area for additional information • date with validation, for example date picker or dd/mm/yyyy or similar • numeric field with validation for party size i.e. max 6 • radio buttons/select element (drop-down) for game theme	**4**	**1 mark** for submit **AND** **1 mark** for each subsequent bullet to a maximum of three. 
`	(c)		• **developer benefit** - changes based on feedback can be made by the developer before development or implementation time has been wasted • **client benefit** - can provide feedback on matching their requirements/fitness for purpose and suggest changes	**2**	**1 mark** for each bullet
	(d)	(i)	`nav ul {list-style-type:none}`	**1**	
		(ii)	• use of the descendant selector • only selects unordered lists within the nav element/doesn't affect other unordered lists not within a nav element	**2**	**1 mark** for each bullet

[END OF MARKING INSTRUCTIONS]