National
Qualifications
2023

2023 Computing Science

Higher

Finalised 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) If a candidate puts a score 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.

(g) 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.

(h) 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.

(i) In the 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.			-16	**1**	
2.			Award 1 mark for each bullet. • Communication/feedback between client and developer occurs throughout the process • Prototypes/software available (for review) • Repeated testing/evaluation • Reduced documentation • Communication within the development team	**2**	
3.	(a)		Award 1 mark for each bullet. • Sign: 1 • Remaining mantissa: 101 0000 1111 0000 • Exponent: 1111 1011	**3**	
	(b)		Floating point numbers could be stored more precisely/accurately.	**1**	
4.			Public key encrypts/encodes	**1**	

Question			Expected response	Max mark	Additional guidance
5.	(a)		Award 1 mark for each bullet. • The address bus is set up with the address (by the processor) • The instruction is transferred on the data bus from memory/to processor	**2**	Must be an indication of direction of travel
	(b)		Award 1 mark for each bullet from factor with matching explanation. Maximum 2 marks. • Clock speed • affects the number of instructions executed/carried out per second **OR** • Data bus width • affects the number of bits/data transferred in a single operation/reduces number of fetch operations **OR** • Cache • affects the number of fetches/ accesses from slower main memory. **OR** • Number of cores • affects how many instructions can be executed concurrently	**2**	

<table>
<tr><th colspan="3">Question</th><th>Expected response</th><th>Max mark</th><th>Additional guidance</th></tr>
<tr><td>6.</td><td></td><td></td><td>Award 1 mark for each bullet.
Maximum 2 marks.

• Unauthorised access
• Intent to commit (a further) offence
• Unauthorised modification</td><td>2</td><td></td></tr>
<tr><td>7.</td><td>(a)</td><td></td><td><code>validPalin (= TRUE)</code></td><td>1</td><td></td></tr>
<tr><td></td><td>(b)</td><td></td><td>• Function <code>checkPalindrome</code> called correctly with single parameter <code>userWord</code>
• Assigning result to <code>palindrome</code></td><td>2</td><td>SQARL:
<code>SET palindrome TO checkPalindrome(userWord)</code></td></tr>
<tr><td>8.</td><td>(a)</td><td></td><td>Award 1 mark for each bullet.

• winningNos[index[10
winningNos[index] 14

• numMatches 1
numMatches 2</td><td>2</td><td>
<table>
<tr><th>Breakpoint</th><th>Variable</th><th>Value</th></tr>
<tr><td rowspan="3">1st Iteration</td><td>index</td><td>0</td></tr>
<tr><td>winningNos[index]</td><td>10</td></tr>
<tr><td>numMatches</td><td>1</td></tr>
<tr><td rowspan="3">2nd Iteration</td><td>index</td><td>1</td></tr>
<tr><td>winningNos[index]</td><td>14</td></tr>
<tr><td>numMatches</td><td>2</td></tr>
</table>
</td></tr>
<tr><td></td><td>(b)</td><td></td><td>If there are no matches to any of the IF conditions the ELSE statement (Line 24/25) is executed.</td><td>1</td><td></td></tr>
</table>

<table>
<tr><th colspan="3">Question</th><th>Expected response</th><th>Max mark</th><th>Additional guidance</th></tr>
<tr><td>9.</td><td>(a)</td><td>(i)</td><td>Award 2 marks for all three correct.
Award 1 mark for any two correct OR three correct with a single column of additional output.

A872
D321
A423</td><td>2</td><td></td></tr>
<tr><td></td><td></td><td>(ii)</td><td>Evaluate towns then visits

OR

Evaluate towns and visits twice</td><td>1</td><td><pre>IF
(city[i] = "Stirling" OR
city[i] = "Dundee")
AND visits[i] > 4 THEN</pre>OR

<pre>IF
(city[i] = "Stirling"
AND visits[i] > 4) OR
(city[i] = "Dundee"
AND visits[i] > 4) THEN</pre>Brackets not required for this expression.</td></tr>
<tr><td></td><td>(b)</td><td></td><td>Award 2 marks for correct pair of parameters.

Award 1 mark for any correct formal/actual parameter.

Award 1 mark for correct pair of parameters wrong way round.

<table><tr><th>Formal</th><th>Actual</th></tr><tr><td>petNo</td><td>dogID</td></tr><tr><td>city</td><td>branch</td></tr><tr><td>visits</td><td>noOfVisits</td></tr></table></td><td>2</td><td></td></tr>
<tr><td></td><td>(c)</td><td></td><td>Award 1 mark for each bullet.

• To call the procedure/subroutine
• And pass the (actual) parameters into that procedure/subroutine</td><td>2</td><td></td></tr>
</table>

Question			Expected response	Max mark	Additional guidance
9.	(d)	(i)	i petNo city visits	**1**	
		(ii)	Award 1 mark for each bullet. Maximum 1 mark. • Only accessible in the (customerSearch) procedure • Accessible across lines 1 to 7 or 2 to 6	**1**	
		(iii)	Award 1 mark for each bullet. Maximum 1 mark. • Editing one subroutine will not affect local variables with the same name in another subroutine • Relies on fewer global variables/ parameters so reduces likelihood of errors	**1**	

	(e)		Award 1 mark for each bullet.
			
			• Initialise and increment counter
			• Use of fixed loop, branch input (and counter output)
			• IF condition matching user input to `branch[]` array
			• And condition `noOfVisits[]` > 5
			
			`DECLARE searchBranch INITIALLY FROM KEYBOARD`
			`DECLARE count INITIALLY 0`
			`FOR index FROM 0 TO length(branch)`
			`If searchBranch = branch[index] and noOfVisits[index] > 5 THEN`
			`  SET count TO count + 1`
			`END IF`
			`SEND count TO DISPLAY`
				4	Accept alternate array names

Question			Expected response	Max mark	Additional guidance
10.	(a)		Award 1 mark for each bullet. Maximum 1 mark. • Only allows characters with ASCII codes. • Password has a minimum of one character	**1**	
	(b)		Award 1 mark for each bullet. • Initialise and update total • Loop for length of password and get ASCII value of current character • Use of mod 11 to get remainder • Concatenate remainder to original password • Open/write to/close file	**5**	
	(c)		Award 1 mark for each bullet. Maximum 2 marks. • Facilitates module libraries which allows the reuse of code • Each part of the solution can be tested independently. • Errors are easier to locate. • Different programmers can code different modules (independently). • Variables only in memory during execution of subroutine. • Limits accessibility of local variables.	**2**	

<table>
<tr><th colspan="3">Question</th><th>Expected response</th><th>Max mark</th><th>Additional guidance</th></tr>
<tr><td>11.</td><td>(a)</td><td>(i)</td><td>Award 1 mark for each bullet.

• Record structure with name <code>feature</code>
• Six fields specified</td><td>2</td><td></td></tr>
<tr><td colspan="6">SQARL

<code>RECORD feature IS (STRING brand, STRING refNo, INTEGER maxWashLoad, INTEGER spinSpeed, REAL price, INTEGER numberInStock)</code></td></tr>
<tr><td></td><td></td><td>(ii)</td><td>Award 1 mark for each bullet.

• Array structure and size (named <code>machines</code>)
• Using <code>feature</code> data type</td><td>2</td><td>SQARL
<code>DECLARE machines AS ARRAY OF feature INITIALLY[]*80</code>
Python
<code>machines = [feature() for x in range(80)]</code>
parameters to initialise record may be passed in
for example
<code>feature('','',...)</code>

VB
<code>Dim machines(79/80) as feature</code></td></tr>
<tr><td></td><td>(b)</td><td></td><td>Award 1 mark for each bullet:

• STEP 3 IN: smallestWash, slowestSpin, machines[ ]

• STEP 4 IN: cheapestPrice, found</td><td>2</td><td></td></tr>
</table>

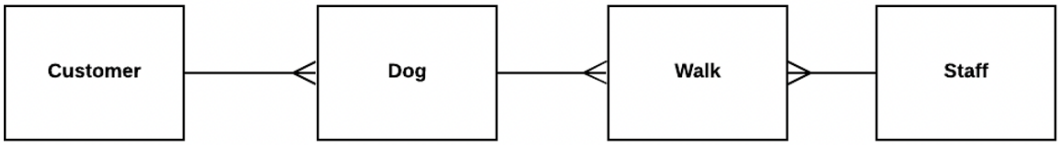
Question			Expected response	Max mark	Additional guidance
11.	(c)		Award 1 mark for each bullet. • Suitable initialisation of cheapestPrice • [ ].washLoad >= smallestWash AND [ ].spinSpeed >= slowestSpin • price < cheapestPrice • Assign cheapest • Found set to false, changed to true when all conditions met • Use of array of records, `machines` • with matching fields from a(i) (dot notation)	**7**	Candidates may initialise cheapestPosition If an iterator is used eg `FOR EACH washer FROM machines DO` an index for the array is not required, for example `washer.spinSpeed`

```
SET cheapestPrice TO 999999
SET found TO FALSE

FOR index FROM 0 TO 79 DO
  IF machines[index].maxWashLoad >= smallestWash AND
     machines[index].spinSpeed >= slowestSpin AND machines[index].price <
     cheapestPrice THEN
          SET cheapestPrice TO machines[index].price
          SET found TO TRUE
  END IF
END FOR
```

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>12.</td><td></td><td></td><td>Award 1 mark for each bullet.

• Customer – Dog (1:M)
• Staff – Walk (1:M)
• Dog – Walk (1:M)</td><td>3</td><td></td></tr>
<tr><td></td><td></td><td></td><td colspan="3"> 
</td></tr>
<tr><td>13.</td><td></td><td></td><td>Award 1 mark for each bullet.

• Grouping on type to return three rows
• Corresponding values for type of car</td><td>2</td><td>Award 0 marks if Saloon is included.
Example: rows can be in any order
<table><tr><th>type</th><th>Cheapest Price</th></tr><tr><td>Hatchback</td><td>15100</td></tr><tr><td>Estate</td><td>13400</td></tr><tr><td>Hybrid</td><td>18200</td></tr></table></td></tr>
</table>

<table>
<tr><th colspan="3">Question</th><th>Expected response</th><th>Max mark</th><th>Additional guidance</th></tr>
<tr><td>14.</td><td>(a)</td><td></td><td>Award 1 mark for each bullet.

• COUNT(any field or *)
• WHERE condition custID =41 and date in May 23

<table><tr><td>Field(s) and Calculation(s)</td><td>COUNT(*)</td></tr><tr><td>Table(s)</td><td>CustOrder</td></tr><tr><td>Search Criteria</td><td>custID = 41 AND orderDate LIKE “%/05/2023”</td></tr><tr><td>Grouping</td><td></td></tr><tr><td>Sort Order</td><td></td></tr></table></td><td>2</td><td>Do not penalise additional entries in Field(s) and Calculation(s).

In design accept other expressions eg
= for LIKE
BETWEEN for dates

SQL allows:

custID = 41 AND orderDate LIKE “%/05/2023”
custID = 41 AND orderDate LIKE “ _ _/05/2023”

Access allows:
custID = 41 AND orderDate LIKE “*/05/2023”
custID = 41 AND orderDate LIKE “??/05/2023”</td></tr>
<tr><td></td><td>(b)</td><td></td><td>Award 1 mark for each bullet.

• Correct three fields
• All four tables</td><td>2</td><td><table><tr><td>Field(s) and Calculation(s)</td><td>forename, surname, telephone</td></tr><tr><td>Table(s)</td><td>Customer, CustOrder, OrderItem, Dish</td></tr><tr><td>Search Criteria</td><td>course = ‘Main’</td></tr><tr><td>Grouping</td><td>(CustID)</td></tr><tr><td>Sort Order</td><td></td></tr></table></td></tr>
<tr><td></td><td>(c)</td><td></td><td>Award 1 mark for each bullet.

• UPDATE Dish
• SET price = price + 2
• WHERE course = ‘Main’ AND description LIKE ‘%Chicken%’</td><td>3</td><td>Award 0 marks for first bullet if SQL clauses are in wrong order

Access:
Accept * for wildcard</td></tr>
<tr><td></td><td>(d)</td><td></td><td>Award 1 mark for each bullet.

• Missing SUM/TOTAL around calculation
• Missing join between OrderItem and Dish tables
• Missing GROUP BY orderID, (orderDate)</td><td>3</td><td></td></tr>
</table>

<table>
<tr><th colspan="3">Question</th><th>Expected response</th><th>Max mark</th><th>Additional guidance</th></tr>
<tr><td>15.</td><td>(a)</td><td></td><td>Award 1 mark for each bullet.

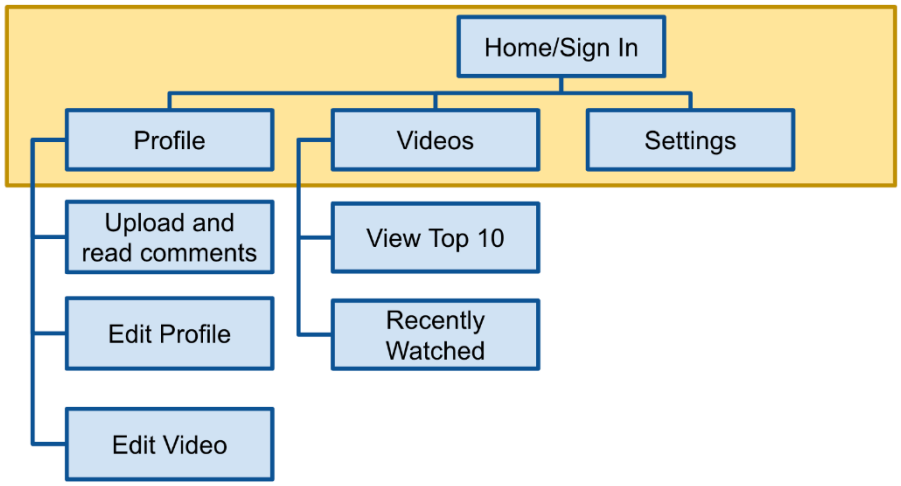
• Fields with calculation (and alias)
• Pilot and Entry tables, where position =1
• Group by pilotID and/or forename, surname</td><td>3</td><td><table><tr><td>Field(s) and Calculation(s)</td><td>pilotID,forename, surname,COUNT(*) *150 (AS ‘Winnings’)</td></tr><tr><td>Table(s)</td><td>Pilot,Entry</td></tr><tr><td>Search Criteria</td><td>position = 1</td></tr><tr><td>Grouping</td><td>pilotID, (forename, surname)</td></tr><tr><td>Sort Order</td><td></td></tr></table></td></tr>
<tr><td></td><td>(b)</td><td></td><td>Award 1 mark for each bullet.

• SELECT with title and AVG(raceTime)
• ALIAS and join for Entry and Race tables
• GROUP BY title or raceID</td><td>3</td><td><pre>SELECT title, AVG(raceTime)
AS 'Average Time'
FROM Entry,Race WHERE
Entry.raceID = Race.raceID
GROUP BY title</pre></td></tr>
<tr><td></td><td>(c)</td><td></td><td>Award 1 mark for each bullet.

• FROM includes Pilot, Entry with Pilot.pilotID = Entry.pilotID in WHERE
• FROM clause includes ‘Fastest’
• WHERE raceTime =[FastestTime]</td><td>3</td><td>Do not award bullet point 1 if a GROUP BY clause is included

<pre>SELECT forename,surname
FROM Pilot, Entry, Fastest
WHERE raceTime=[FastestTime]
AND Pilot.pilotID =
Entry.pilotID</pre>Sub query also acceptable:
<pre>SELECT forename, surname
FROM Pilot,Entry
WHERE racetime = (SELECT
MIN(racetime) FROM Entry)
and Entry.PilotID =
Pilot.pilotID</pre></td></tr>
<tr><td></td><td>(d)</td><td></td><td>The primary key/raceID and pilotID/compound key would no longer be unique</td><td>1</td><td></td></tr>
</table>

Section 3 – Web design and development

Question			Expected response	Max mark	Additional guidance
16.			Award 1 mark for each bullet. • Does not change to another image • Function not called/no onClick event/code	**2**	
17.	(a)		Award 1 mark for each bullet. Maximum 1 mark. • A form to: add/read comments user sign in edit profiles/video details • An element to upload videos watch videos	**1**	
	(b)		Award 1 mark for each bullet. • All pages in level 1 • All pages in level 2  	**2**	

Question			Expected response	Max mark	Additional guidance
18.	(a)		Award 1 mark for each bullet. • Grouping h2,h3,img • Correct margins: 10 px	**2**	`h2,h3,img {margin:10px;}`
	(b)		float: left; **OR** display: inline;	**1**	Accept inline-block;
	(c)	(i)	Award 1 mark for each bullet. • ID: #margInfo • display: none	**2**	`#margInfo {display:none;}` Accept inline style:
		(ii)	block inline	**1**	Accept inline-block;
		(iii)	Award 1 mark for each bullet. • Image added • Onmouseover event with function	**2**	`<img src=”mPizza.jpg” onmouseover="displayMText() ">` May include (this) as parameter to the function
	(d)	(i)	`<input type=”text” …>`	**1**	
		(ii)	Award 1 mark for each bullet. • <textarea> (with closing </textarea>) • required attribute within an element	**2**	<textarea required></textarea> Do not accept <input type=”text”>

Question			Expected response	Max mark	Additional guidance
19.	(a)		Award 1 mark for each bullet. • Labelled fields for full name, email, date of birth, work experience • Validated selection for most recent education • Submit button	**3**	Accept use of radio buttons or select as validation.
	(b)	(i)	Award 1 mark for each bullet. • list-style-type: none will remove the bullet points • li a color: grey overwrites the li color: black styling so can't read grey text on the grey background (until you hover over the text)	**2**	
		(ii)	Award 1 mark for each bullet. • CSS Rule 1 changes the (colour of text of) anchor/links in a list (to white when hovered over) • CSS Rule 2 changes the (colour of text of) every anchor/link (to yellow when hovered over)	**2**	
	(c)		Award 1 mark for each bullet. • Different device/screen sizes/resolution would make layouts different • Different browsers (some colours styling would look different)	**2**	

[END OF MARKING INSTRUCTIONS]