National
Qualifications
2024

2024 Computing Science

Higher

Question Paper Finalised Marking Instructions

These marking instructions have been prepared by examination teams for use by SQA appointed markers when marking external course assessments.

SQA ©

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 in brackets() 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)		1110 0111	**1**	
	(b)		$2^7 - 1$	**1**	Also accept • 127 • 0111 1111 • (-128) + 127
2.			• Agile – evaluation will be ongoing throughout the development process • Iterative – evaluation will take place at the end of the process/after testing	**2**	
3.	(a)		• Sign bit: 1 • Remaining mantissa: 111 0000 0000 0000 • Exponent: 0000 0000	**3**	
	(b)		Range will be increased	**1**	

Question			Expected response	Max mark	Additional guidance
4.			• Initialise maxLength to 0/length of first surname/position = 0 • Loop until the end of surnames • If length of current surname>maxLength/name at max position then • Set maxLength to length of current surname/maxposition to current position	**4**	
5.			• Cache has faster access time than main memory • Stores frequently used data/instructions • Cache hits can occur • Reducing the need to access/fetch execute from main memory	**2**	Award 1 mark for each bullet Maximum 2 marks

6.	(a)		• Use of mod function/operator
			• Arguments (n, divisor) in order with comparison to 0
				2	Example answers:
					`n Mod divisor =0`
					
					Python (and Java)
					`accept single =`
					`n % divisor == 0`
					
					Accept single = symbol
					VB
					`n Mod divisor = 0`

	(b)		• Use of function name with actual parameter `inputNum`
			• Assignment to `isPrime`
				2	Example answers:
					
					`SET isPrime TO checkPrime(inputNum)`
					
					`isPrime=checkPrime(inputNum)`

<table>
<tr><th colspan="3">Question</th><th>Expected response</th><th>Max mark</th><th>Additional guidance</th></tr>
<tr><td>7.</td><td>(a)</td><td></td><td>• array of 750 elements.
• use of record structure (<code>Recipe</code>)</td><td>2</td><td>Allow indexing from 1 to 750.

Example answers:

<code>DECLARE allRecipes AS ARRAY OF Recipe[]*750</code>

<code>DECLARE allRecipes AS ARRAY OF Recipe INITIALLY[NULL]*750</code>

Award 1 mark for
<code>allRecipes=[Recipe()]*750</code>
<code>allRecipes=[Recipe]*750</code></td></tr>
<tr><td colspan="6">Python <code>allRecipes = [Recipe() for x in range(750)]</code>
VB <code>Dim allRecipes(749) as Recipe (accept 750)</code>
Java <code>Recipe[] allRecipes = new Recipe[750];</code></td></tr>
<tr><td></td><td>(b)</td><td></td><td>• initialisation of found flag/result and update
• condition = search ingredient and <= searchTime using an array
• array name matching to (a)
• use of field names in IF condition (ingredient,minutes)
• display message with two values inside loop that traverses entire array
• single output of no match displayed after the loop</td><td>6</td><td>If incorrect field names are used, only penalise once for bullets 2,4 and 5

If parallel arrays are used do not award bullets 3 and 4

At bullet 2 do not penalise if parallel arrays are used</td></tr>
<tr><td colspan="6">Example answer:

<pre>SET Found TO False
RECEIVE searchIngred FROM KEYBOARD
RECEIVE searchTime FROM KEYBOARD

FOR n FROM 0 TO LEN(allRecipes) DO
   IF allRecipes[n].ingredient = searchIngred AND
      allRecipes[n].minutes <= searchTime THEN
        SEND allRecipes[n].title &" requires " & allRecipes[n].minutes & "
        minutes." TO DISPLAY
      Found = True
   END IF
END FOR
IF Found =FALSE
   SEND "No matches for that search, try again." TO DISPLAY
END IF</pre></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>8.</td><td>(a)</td><td></td><td>• All inputs must be integers
• Number of values must be <= maximum value – minimum value
• Maximum value > minimum value /minimum value < maximum value
• Number of values >0</td><td>2</td><td>Award 1 mark for each bullet. Maximum 2 marks

Do not award marks that refer to the output or generation of random numbers</td></tr>
<tr><td></td><td>(b)</td><td>(i)</td><td>Line 35 valid should be set to TRUE

OR

Function always returns valid as FALSE</td><td>1</td><td>Valid is initially set to false and, in the decision can only be set to false again, meaning it can never be set to true</td></tr>
<tr><td></td><td></td><td>(ii)</td><td>• While condition at 66 will always be met/valid is always FALSE
• Infinite loop</td><td>2</td><td></td></tr>
<tr><td></td><td>(c)</td><td></td><td>• Use conditional loop
• To exit as soon as the number is present in the array/<code>randomNum</code> is present in <code>values</code>/array OR valid = False</td><td>2</td><td></td></tr>
<tr><td></td><td>(d)</td><td></td><td><table><tr><td>Formal</td><td>randomNum</td><td>values</td></tr><tr><td>Actual</td><td>randomVal</td><td>randomList</td></tr></table></td><td>2</td><td>Award 2 marks for correct pair

Award 1 mark for:
• Correct pair of parameters the wrong way round

OR

• Two formal or two actual parameters</td></tr>
</table>

Question			Expected response	Max mark	Additional guidance
9.	(a)		STEP 2 IN: distance[],drivingTime[] STEP 2 OUT: avgSpeed[] STEP 3 IN: distance[],drivingTime[] STEP 4 IN: avgSpeed[],distance[],avgDistance	**4**	Award 1 mark for each step STEP 2 OUT: must indicate an array, shape of brackets is not relevant If avgSpeed variable is passed out of Step 2 without brackets do not penalise again at Step 4 IN If candidate has entered any data flow for Step 1 IN or Step 4 OUT do not award marks for that Step
	(b)		Identifies parameters/variables	**1**	
	(c)		• Initialisation of journeys and total distance, increment of number of journeys • Loop and update of total distance • If statement with correct condition • Correct calculation of average	**4**	
	(d)	(i)	System resources such as processor/backing storage/RAM are exhausted/fully utilised	**1**	
		(ii)	• Loss of revenue • Repair (of damage due to attack) • Prevention of future attacks	**1**	Maintenance would be ongoing and not solely as a result of a DOS attack

<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>(a)</td><td></td><td>• Line 15 counter = 1
• Line 12 index = 1 AND Line 13 counter = 0
• Line 18 longestStreak = 0 AND Line 20 counter = 0

Example answer:

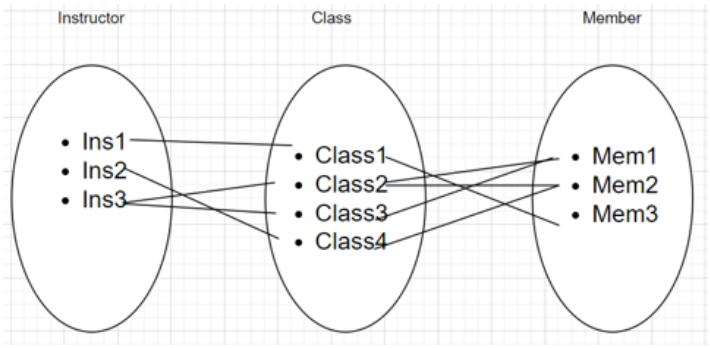
<table><tr><th>Line Number</th><th>counter</th><th>index</th><th>longestStreak</th></tr><tr><td>10</td><td>0</td><td></td><td></td></tr><tr><td>11</td><td></td><td></td><td>-1</td></tr><tr><td>12</td><td></td><td>0</td><td></td></tr><tr><td>13</td><td>0</td><td></td><td></td></tr><tr><td>15</td><td>1</td><td></td><td></td></tr><tr><td>12</td><td></td><td>1</td><td></td></tr><tr><td>13</td><td>0</td><td></td><td></td></tr><tr><td>18</td><td></td><td></td><td>0</td></tr><tr><td>20</td><td>0</td><td></td><td></td></tr></table></td><td>3</td><td>Do not penalise if candidate records all values at each line number</td></tr>
<tr><td></td><td>(b)</td><td></td><td>Line 13</td><td>1</td><td></td></tr>
<tr><td></td><td>(c)</td><td></td><td>• Breakpoint
• Stops execution at a specified line/allowing variable values to be compared to expected values/allows variables to be inspected

OR

• Watchpoint
• Stops execution when a variable changes or meets condition/allowing variable values to be compared to expected values/allows variable to be inspected</td><td>2</td><td>Award 1 mark for correct description of dry run</td></tr>
<tr><td></td><td>(d)</td><td></td><td>It is declared/only accessible or used inside the function (lines 9-27)</td><td>1</td><td></td></tr>
</table>

Question			Expected response	Max mark	Additional guidance
10.	(e)		Efficiency • The function can be called more than once/reused (with different parameters) • Local variables/parameters only held in memory while being executed Maintainability • If there is an error with that part of the program then the bug is easier to find/easier to identify errors in a section of code • Is increased because modules with local variables can be edited without affecting the other modules in the program • Any new modules or changes to existing modules can be edited/tested individually	**2**	Award 1 mark for any one bullet about efficiency Award 1 mark for any one bullet about maintainability

Section 2 – Database design and development

Question			Expected response	Max mark	Additional guidance
11.	(a)	(i)	• Create a query to COUNT /calculate the number of classes run by each instructor • Create a query to COUNT/find MAX/search for the member(s) who attends the greatest number of classes • Create a query to COUNT/search for the member(s) who attend more than 5 classes • Create a query to SUM/calculate the total cost of a member's classes	**1**	Award 1 mark for any one bullet
		(ii)	• COUNT (bullet 1 or 3) • COUNT/MAX (bullet 2) • SUM (bullet 4)	**1**	Award 1 mark for any one bullet Function stated must match to the requirement identified in part (i)
	(b)		• Entity and instance names • Correct associations between instances Example answer:  	**2**	Diagram may be reversed
12.			• tournamentID is part of the compound/primary key • A primary key cannot be blank/null	**2**	

<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>• Correct UPDATE syntax with SET and WHERE in correct order with suitable condition
• Correct calculation</td><td>2</td><td>Example answer:

<pre>UPDATE Item
SET quantity = quantity + 20
WHERE itemName = "Orange"</pre>
Allow:
<pre>WHERE itemid = 151</pre></td></tr>
<tr><td></td><td>(b)</td><td></td><td>• DELETE FROM Supplier with WHERE supplierRef
• LIKE with wildcard after P</td><td>2</td><td>Example answer:

<pre>DELETE FROM Supplier
WHERE supplierRef LIKE "P%"</pre>
Allow Access wildcard *</td></tr>
<tr><td></td><td>(c)</td><td></td><td>• Type field and calculation with alias
• MAX function
• Table and grouping
• Sort using calculation with function/alias DESC

Example answer:
<table><tr><td>Field(s) and calculation(s)</td><td>type, MAX(sellingPrice-buyingPrice) as Profit</td></tr><tr><td>Table(s)</td><td>Item</td></tr><tr><td>Search criteria</td><td></td></tr><tr><td>Grouping</td><td>type</td></tr><tr><td>Sort Order</td><td>MAX(sellingPrice-buyingPrice) OR Profit DESC</td></tr></table></td><td>4</td><td>Do not award bullet 1 if additional fields in Fields & Calculations

Do not double penalise in bullet 4 if incorrect calculation (from bullet 1) is repeated

Ignore fruit or veg in search criteria</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>• Tables(Instructor, Pupil, Booking)
• Condition hourlyRate > 35
• Equi Joins Instructor.instructorID = Booking.instructorID AND Booking.pupilRef = Pupil.pupilRef</td><td>3</td><td>Example answer:

<code>FROM Instructor, Pupil, Booking
WHERE hourlyRate > 35
AND Instructor.instructorID = Booking.instructorID AND Booking.pupilRef = Pupil.pupilRef</code>

Award bullet 3 if a NATURAL join is used without field names</td></tr>
<tr><td></td><td>(b)</td><td>(i)</td><td>• Totals
• Doubly sorted output</td><td>2</td><td>Example answer:
<table><tr><th>town</th><th>Number Per Town</th></tr><tr><td>Greenock</td><td>3</td></tr><tr><td>Falkirk</td><td>2</td></tr><tr><td>Kilmarnock</td><td>2</td></tr><tr><td>Airdrie</td><td>1</td></tr></table></td></tr>
<tr><td></td><td></td><td>(ii)</td><td>• Produces a single output for each town
• Without a GROUP BY it would produce a single output for only one town
• In the SELECT clause the town field is a non-aggregate field</td><td>1</td><td>Award 1 mark for any one bullet</td></tr>
<tr><td></td><td>(c)</td><td></td><td>• AVG function on hourlyRate
• Alias and table
• Criteria for dayOff</td><td>3</td><td>Example answer:

<code>SELECT AVG(hourlyRate) as [Average Hourly Rate]
FROM instructor
WHERE dayOff = “Saturday” OR dayOff=”Sunday”

WHERE dayOFF LIKE “S%”</code>
Access:
<code>dayOff LIKE “S*”</code></td></tr>
<tr><td></td><td>(d)</td><td></td><td>• First query to identify minimum/cheapest hourly rate
• First query to be included in a second query (using query name in the FROM and alias in WHERE clause)

OR

• Use a subquery to identify minimum/cheapest hourly rate (1 mark
• Within the WHERE clause</td><td>2</td><td>Candidate can create a view which saves the results of a query similar to a table in the FROM clause</td></tr>
</table>

Section 3 – Web design and development

Question			Expected response	Max mark	Additional guidance
15.			• `main, section {background-color: red}` • `section, p, h1, h2 {padding: 5px}` • `h1 {color: white; font-size: 22px}`	**3**	
16.	(a)		Personas are fictitious users created to accurately represent the users of the website	**1**	
	(b)		Try to log in • with an incorrect detail • with account details that are not registered • with valid details Try to create an account • with all details/valid password length • leaving some of the details blank • with a password that doesn't meet the credentials	**2**	Award 1 mark for each bullet Maximum 2 marks

Question			Expected response	Max mark	Additional guidance
17.	(a)	(i)	There are links to pages that should not be there (School Uniform and School Day) **OR** Links to School Uniform/School day should be on Parents/Carers page	**1**	

		(ii)	• `list-style-type`
			• `float:left`
			• `a:hover`
				3	`nav ul {`**`list-style-type`**`:none}`
					`nav ul li {`**`float:left`**`;width:180px}`
					`nav ul li a {display:block;padding:6px}`
					`nav ul li `**`a:hover`**
					`{background-color:white;color:black}`

	(b)	(i)	`<img src="mon.jpg" onmouseover="displayMon()">`	**1**	Accept `displayMon(this)` Must have brackets for function call
		(ii)	A: hideAllDays(); B: block/inline	**2**	Accept `hideAllDays(this)` Must have brackets for function call

Question			Expected response	Max mark	Additional guidance

18.	(a)		• left margin (applied to #flightSimulator) AND vertical margin (applied to bottom of flight simulator or top of #giftVouchers)
			• height/width to #giftVouchers
			• clear:both to #giftVouchers
				3	`#flightSimulator {`
					`margin-left:10px; margin-bottom:10px;`
					`}`
					
					`#giftVouchers{`
					`width:790px;height:120px; clear:both;` `}`

	(b)	(i)	• Name, contact telephone number, email indicating required • Location and date as suitable drop down, radio buttons etc • Submit button	**3**	
		(ii)	• minlength • required	**2**	
	(c)		• Descendant • Means styling only applied to p elements in the footer	**2**	
	(d)		• Displays/functions as intended on different browsers • Displays/functions as intended on different devices	**2**	

[END OF MARKING INSTRUCTIONS]