National
Qualifications
2026

2026 Computing Science

Advanced Higher

Question Paper Finalised Marking Instructions

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

Qualifications Scotland
Teisteanasan Alba

General marking principles for Advanced 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 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.

(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 detailed 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

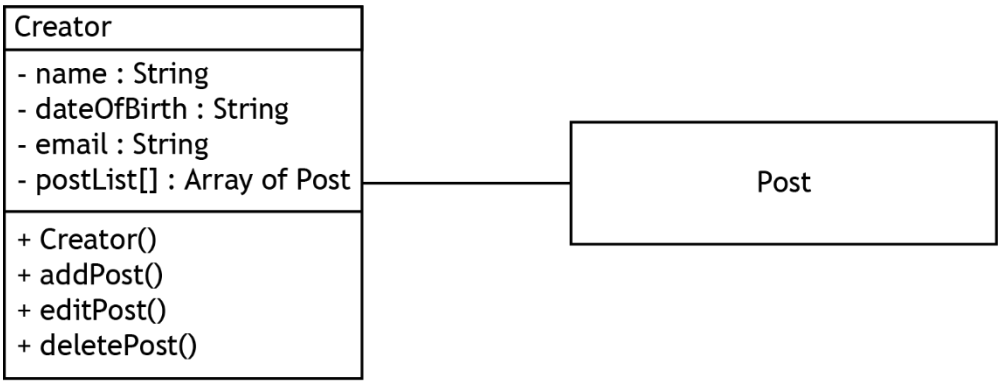
Section 1 — Software design and development

Question			Expected response	Max mark	Additional guidance
1.	(a)		Arrays have a fixed size, which must be defined at the time of creation whereas linked lists allow dynamic memory allocation, meaning they can grow or shrink in size as needed. An Array is a collection of elements stored in contiguous memory locations which allows random access to elements using an index; a linked list is a collection of nodes where each node contains a data element and a reference (or pointer) to the next node in the sequence.	**1**	Award **1 mark** for either comparison point. Must be a valid comparison.
	(b)		See sample solution below. 5. Compare 14 and 19 → no swap → [12, 2, 5, 7, 14, 19, 15, 5, 7] 6. Compare 19 and 15 → swap → [12, 2, 5, 7, 14, 15, 19, 5, 7] 7. Compare 19 and 5 → swap → [12, 2, 5, 7, 14, 15, 5, 19, 7] 8. Compare 19 and 7 → swap → [12, 2, 5, 7, 14, 15, 5, 7, 19]	**2**	Award **1 mark** for steps 5 & 6 correct. Award **1 mark** for steps 7 & 8 correct.
2.	(a)		• **Economic feasibility:** Is there a financial benefit in spending money to create the app. Will the benefits of developing the app outweigh the development costs • **Technical feasibility:** Is there expertise amongst the developers to create the app. What operating system should the app be compatible with	**2**	Award **1 mark** for economic. Award **1 mark** for technical.
	(b)		• the developers would adopt the characteristics of the persona • they would use the screen reader and speech-to-text facilities of a mobile phone to use the app • each of the planned test cases would be carried out by the developers to enable them to consider the suitability of the app for this user	**2**	Award **1 mark** for either of the following: • developer adopting characteristics of the persona • using the screen reader speech-to-text facility Award **1 mark** for carrying out each of the test cases.

Question			Expected response	Max mark	Additional guidance
3.	(a)	(i)	• this statement creates an instance of the `Post` class called `post1` • the constructor of the `Post` class then assigns the following values to each of the instance variables of the `post1` object: Creator = Ally Text = Loving this beautiful day at the beach Likes = 0 Shares = 0 Comments = null	**2**	Award **1 mark** for instantiation of the object post1 of the `Post` class. Award **1 mark** for the correct assignment of all instance variables.
		(ii)	• post by Ally text: Loving this beautiful day at the beach • likes: 3 shares: 2 • comments: No user comments yet	**3**	Award **1 mark** for the correct row headings. Award **1 mark** for the correct number of likes and shares. Award **1 mark** for No user comments yet.

	(b)	(i)	See sample solution below.
			
			`OVERRIDE FUNCTION countInteractions() RETURNS INTEGER`
			`  DECLARE interactions INITIALLY 0`
			`  SET interactions TO THIS.likes + THIS.shares + length (THIS.comments) + THIS.views`
			`  RETURN interactions`
			`END FUNCTION`
				1	Award **1 mark** for correct calculation as part of a function.

		(ii)	• `posts` is an array of objects of type `Post`. Due to polymorphism, `posts` can contain objects of type `Post`, `Image` or `VideoPost` • `posts[1]` points to variable `post2` which is an instance of the `VideoPost` class, however the array `posts` has been defined as type `Post`. As a result, the only methods that can be accessed are those in the `Post` superclass • the `addView()` method is only associated with the `VideoPost` subclass so an error will be generated when any of the array elements try to access it	**2**	Award **1 mark** for each bullet. Maximum **2 marks**.

Question			Expected response	Max mark	Additional guidance
3.	(c)		• A: 834 < 1228 • B: Object at index 0 copied to index 1 • C: Temp object inserted at index 2	**3**	
	(d)		See sample solution below.  	**2**	Award **1 mark** for correct properties, types and encapsulation. Award **1 mark** for correct methods, including constructor.
4.	(a)	(i)	The developer can make changes to the requirements based on user opinions. The developer can incorporate user suggestions in the game.	**1**	Award **1 mark** for benefit for developers.
		(ii)	Users can give feedback on features they'd like to see in the new game.	**1**	Award **1 mark** for benefit for end users.
	(b)		See sample solution below. `DECLARE gameboard AS ARRAY OF ARRAY OF Square < 9 x 7 array with all elements initialised as an empty record >`	**1**	Award **1 mark** for 2D array with correct dimensions and type Square

Question			Expected response	Max mark	Additional guidance

4.	(c)	(i)	Solutions below are based on row 0 at top of the game board.
			
			Square 37
			`SET gameboard[3][5].pointer TO 0`
			
			Square 44
			`SET gameboard[2][1].pointer TO 30`
				1	Award **1 mark** for correct code used to assign both pointer values correctly.
					
					Accept solutions based on row 0 at bottom of the game board, eg
					Square 37
					`SET gameboard[5][5].pointer TO 0`
					Square 44
					`SET gameboard[6][1].pointer TO 30`
					
					Also accept solutions with row and column values swapped throughout, eg
					Square 37
					`SET gameboard[5][3].pointer TO 0` or
					`SET gameboard[5][5].pointer TO 0`
					
					Square 44
					`SET gameboard[1][2].pointer TO 30` or
					`SET gameboard[1][6].pointer TO 30`

		(ii)	See sample solution below.	**2**	Award **1 mark** for correct use of 2D array and record structure for assignment of position numbers. Award **1 mark** for assignment of continuous position numbers, taking account of odd and even rows. Accept solution based on row 0 at bottom of the game board.

```
1.  set posNumber = 1
2.  loop for each row from 8 to 0
3.      if row is even number
4.          loop for each column from 0 to 6
5.              set gameboard[row][column]. position = posNumber
6.              set posNumber = posNumber + 1
7.          end loop
8.      else
9.          for each column from 6 to 0:
10.             set gameboard[row][column].position = posNumber
11.             set posNumber = posNumber + 1
12.         end loop
13.     end if
14. end loop
```

Question			Expected response	Max mark	Additional guidance
4.	(d)	(i)	See possible solution below.	**3**	Award **1 mark** for searching `gameboard` array to determine current position of player `colour` Award **1 mark** for working out value of `newPosition` (step 4). Award **1 mark** for appropriate use made of 2-D array and record data structure.

```
1.  loop for row from 0 to 8
2.      loop for column from 0 to 6
3.          if gameboard[row][column].player = colour then
4.              set newPosition = gameboard[row][column].position + move
5.          end if
6.      end loop
7.  end loop
```

		(ii)	See sample solution below.	**2**	Award **1 mark** for checking for a win and suitable action (record winner colour/ set winner to true/ display winner message (steps 1 & 2). Award **1 mark** for processing bounce back (steps 4 to 5).

```
1.  if newPosition = 63 then
2.      set winner to colour
3.  else
4.      if newPosition > 63 then
5.          set newPosition = 63 – (newPosition – 63)
6.      end if
7.  end if
```

Question			Expected response	Max mark	Additional guidance
4.	(d)	(iii)	See sample solution below. 1. if gameboard[row][column].pointer > 0 then 2. set newPosition = gameboard[row][column].pointer 3. end if	**2**	Award **1 mark** for correct condition to check for ladder or snake (step 1). Award **1 mark** for correctly updating `newPosition` using stored pointer value (step 2).
	(e)		• integrative testing would be needed for this program because it consists of several separate modules such as rules12, rules34 and rules56 which will be developed separately • it would be used to ensure that interaction between the main program and individual modules is carried out successfully and that there are no conflicts	**2**	Award **1 mark** for each module being developed separately. Award **1 mark** for discussion of interaction between the modules or possible conflicts.

Section 2 — Database design and development

Question			Expected response	Max mark	Additional guidance
5.	(a)		• there is no relationship between the Administrator and Customer actors to show generalisation • cancel booking should be an extend use case from view booking history • cancel booking has not been connected to the database	**2**	Award **1 mark** for valid response. Maximum **2 marks**.
	(b)	(i)	The Performer entity can exist with no reliance on other entities so is a strong entity.	**1**	Award **1 mark** for responses that indicate that Performer entity is independent of all other entities in the system.
		(ii)	The Concert entity needs a performerID from the Performer entity to exist which makes it a weak entity.	**1**	Award **1 mark** for responses that indicate that the Concert entity can't exist without having a performerID.
		(iii)	A surrogate key would replace the compound key which would make it easier to form joins with the Customer and Concert entities and simplyfying query code. A single surrogate key would make it easier to access historical data. Queries would execute more quickly because joins would be formed using a single value.	**1**	Award **1 mark** for description of one benefit that would be gained from use of a surrogate key.
	(c)		• A: concertDate BETWEEN 1st May 2026 AND 30th June 2026 • B: HAVING C: count(*) >=5	**2**	Award **1 mark** for A. Award **1 mark** for B and C. Note: Ignore date format used in response.

Question			Expected response	Max mark	Additional guidance
6.	(a)		Possible answers include: • use of varchar(200) for the email field ensures no overly long or inappropriate data is entered • use of varchar(4) for the PK memberID would ensure that the values stored are meaningful but would require more memory than equivalent 4 digit integer values • use of varchar for the PK memberID may be more difficult to keep track than use of integer data type that can be auto-incremented • indicating data types and sizes used means that appropriate test data values can be generated to ensure that, for example, only valid date values can be entered and excessively long memberIDs would be rejected • use of varchar for several fields allows variable-length string values to be stored with no overheads or padding. This means that only the characters entered (up to the stated maximum number of characters) are stored and use of memory is efficient since there is no need to store unnecessary characters • efficient use of the data type varchar will impact on database performance by improving query efficiency and reducing processing time	**2**	Award **1 mark** each for any two relevant points that refer to Implementation and/or Testing.

	(b)		`SELECT className, category`
			`FROM Class`
			`WHERE classID NOT IN (`
			`  SELECT classID`
			`  FROM Enrolment`
			`);`
				2	Award **1 mark** for correct use of `IN` with `classID` and `NOT`
					
					Award 1 mark for subquery with `classID`
					
					NOTE: Do not penalise if fields used in the outer `SELECT` clause are not an exact match for those in the solution.

Question			Expected response	Max mark	Additional guidance
6.	(c)		Query 1 (using ANY) • the subquery executes and returns a list of all `classID`s found in the `Class` table • the outer query then checks each `Enrolment.classID` to see if it matches any value in that list; if it does, the corresponding `memberID` is returned • this means that the subquery must execute in full and the outer query must then perform a full scan of the set of results returned by the subquery Query 2 (using EXISTS) • the subquery is executed for each row of the `Enrolment` table to check whether the subquery returns a `true` or `false` value a value of `true` indicates that the `Enrolment.classID` of the current row has been matched with a `classID` from the `Class` table; when this is the case, the corresponding `memberID` is returned • this means that the `EXISTS` comparison stops evaluating the subquery as soon as a suitable match is found	**3**	Award **1 mark** for referring to results returned by each subquery. Award **1 mark** for referring to how results of subquery are processed by the outer query with the `ANY` operator. Award **1 mark** for referring to how results of subquery are processed by the outer query with the `EXISTS` operator.
	(d)	(i)	`DROP TABLE Member;`	**1**	
		(ii)	`Enrolment` has mandatory participation in its relationship with `Member`. As a result, deletion of the `Member` would not be permitted.	**1**	Note: response must describe the impact of the SQL statement and must refer to relationship participation of `Member` and `Enrolment`
	(e)	(i)	• `GET` is suitable as the data submitted would not be sensitive data • using `GET` allows the search to be bookmarked for future access	**1**	Award **1 mark** for either bullet point.
		(ii)	• use `minlength` and `maxlength` attributes to add a length check • use `SELECT` instead of `INPUT` to generate a drop down list which would restrict the choices available	**1**	Award **1 mark** for accurate description of any one way to improve the input validation. Note: Range check is not appropriate in this situation.

Question			Expected response	Max mark	Additional guidance
6.	(f)		• corrective maintenance • since the search function is returning inaccurate results, this needs to be investigated and errors found need to be fixed	**2**	Award **1 mark** for corrective maintenance. Award **1 mark** for explanation.

Section 3 — Web design and development

Question			Expected response	Max mark	Additional guidance
7.	(a)		• there is no relationship between the Administrator and User actors to show generalisation • cancel Appointment should be an extend use case from View appointments • cancel Appointment has not been connected to the Database	**2**	Award **1 mark** for each bullet. Maximum **2 marks**.
	(b)		See sample solution below. `@media screen and (max-width: 600px) {  }`	**1**	Award **1 mark** for use of media screen query with `max-width` of 600px.
	(c)		• Label A: Hyperlink – display password reset page • Label B: Check for @	**2**	Award **1 mark** each.
	(d)		A session variable has been used to store the email submitted at login so that it can persist across all pages of the website.	**1**	Award **1 mark** for identifying need for email to persist across pages of the website.
8.	(a)		Connection parameters database server, and database are in the wrong order. See correct code below. `mysqli_connect("pawsomeServer", "PetStylist", "pa$$word!23", "petStyling");`	**1**	Award **1 mark** for correct sequence of connection parameters.
	(b)	(i)	See sample solution below. `$results = mysqli_query($connection, $query) or die("Connection Failure");`	**2**	Award **1 mark** for correct use of `mysqli_query()` function with correct use of `$results`, `$query` and `$connection` variables. Award **1 mark** for correct use of `die()` function.

Question			Expected response	Max mark	Additional guidance
8.	(b)	(ii)	See sample solution below. Alternative solutions are possible.	**2**	Award **1 mark** for display of appointment time with name of pet (traversing search results with step 7). Note: apptDate should not be displayed. Award **1 mark** for comparing results returned with times to be displayed to display – when no pet has been booked

1. set $hours = [“9am”, “10am”, “11am”, “12pm”, “1pm”, “2pm”]
2. set count = numRows($results)
3. loop for time from 0 to 5
4. set found = false
5. loop for appointments from 0 to count-1
6. if apptTime of current $results entry = $hours(time) then
7. display apptTime of current $results entry and pet name of petID
8. set found = true
9. end if
10. end loop
11. if found = false then
12. display apptTime of current $results entry and “-”
13. end if
14. end loop

	(c)		See sample solution below.	**2**	Award **1 mark** for correct use of HTML `SELECT` with name and value attributes. Award **1 mark** for correct use of `$ratings` to populate options.

```
echo '<select name = "behaviourRating">';
foreach ($ratings as $rate) {
    echo '<option value = "'.($rate).'">'.($rate).'</option>';
}
echo '</select>';
```

Alternative solution

```
echo '<select name = "behaviourRating">';
for ($i=0; $i<2; $i++) {
    echo '<option value =
"'.($ratings(i)).'">'.($ratings(i)).'</option>';
}
echo '</select>';
```

Question			Expected response	Max mark	Additional guidance
8.	(d)	(i)	See sample solution below.	3	Award **1 mark** for loop with numRows to process rows of `$allPets` but only processing details of pets that have not yet been banned. Award **1 mark** for indicating output messages for all 3 conditions. Award **1 mark** for indicating correct use of update queries for all 3 conditions.
			1. loop for (number of rows in $allPets –1) 2. if current pet not banned then 3. if greenRatings of current pet = 10 then 4. display pet details with discount message 5. use an update query to set value of greenRatings = 0 6. else if yellowRatings of current pet = 3 then 7. display pet details with surcharge message 8. use an update query to set value of yellowRatings field = 0 9. else if redRatings of current pet = 2 then 10. display pet details with banned message 11. use an update query to set value of banned field = true 12. end if 13. end if 14. end loop		
		(ii)	`SELECT COUNT(*) FROM PET WHERE banned = TRUE;`	2	Award **1 mark** for use of `SELECT` query with `COUNT(*)` Award **1 mark** for correct condition.
	(e)		**Perfective maintenance** • this is an additional feature that was not a requirement in the original software specification • system is working so there are no errors that require corrective maintenance to be carried out	2	Award **1 mark** for perfective maintenance. Award **1 mark** for any valid reason.

[END OF MARKING INSTRUCTIONS]