Higher
coursework
assessment task

Qualifications Scotland
Teisteanasan Alba

Higher Computing Science
Assignment
Finalised Marking Instructions

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

The information in this document may be reproduced in support of Qualifications Scotland qualifications only on a non-commercial basis. If it is reproduced, Qualifications Scotland must be clearly acknowledged as the source. If it is to be reproduced for any other purpose, written permission must be obtained from Qualifications Scotland.

© Qualifications Scotland 2026

Marking instructions

General marking principles

Always apply these general principles. Use them in conjunction with the specific marking instructions, which identify the key features required in candidates' responses.

a Always use positive marking. This means candidates accumulate marks for the demonstration of relevant skills, knowledge and understanding; marks are not deducted for errors or omissions.

b If a candidate response does not seem to be covered by either the principles or detailed/specific marking instructions, and you are uncertain how to assess it, you must seek guidance from your team leader.

c Award marks regardless of spelling, as long as the meaning is unambiguous and does not result in a syntax error in implemented code.

d For design and implementation tasks, a sample response may be 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.

e If a candidate puts a score through their entire response to a question 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.

f 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.

Task 1 — software design and development

Task	Expected response	Max mark	Additional guidance
1a	**Input** ♦ read tool, manufacturer, dateRented, returned, fee from file **Process** ♦ search for tools in 2025 that have not been returned ♦ search for (tools from) chosen manufacturer ♦ count of tools by chosen manufacturer	**3**	Award **1 mark** for input. Award **1 mark** each for any two processes. For process bullet one - 2025 required as file also has 2026 data.
1b	♦ **1 mark** for: ○ loop for each tool (120) ○ condition for tool not being returned and ○ use of year and month to correctly set fee ♦ identification of month and year from date ♦ fee array returned outside loop	**3**	Award bullet two if candidate uses entire date.

Task	Expected response	Max mark	Additional guidance
1c	**Read in Tool data (2 marks)** ♦ module with parameters passed or returned ♦ assigned to 5 parallel arrays **Manufacturer search (4 marks)** ♦ module with correct parameters passed in (tool, manufacturer), nothing returned ♦ input of manufacturer and display of count and tool name(s) ♦ count initialised and incremented ♦ search for selected manufacturer (loop with if) **Calculate late fee (4 marks)** ♦ function with correct parameters passed in (dateRented, returned, fee), fee returned ♦ loop with condition ○ loop for each tool (120) ○ condition for returned = no ♦ correct use of sub-string (year and month) within condition ♦ assignment of fee **Write file (4 marks)** ♦ module with correct parameters passed in (tool, dateRented, fee), nothing returned ♦ create and open file ♦ check for late fee ♦ write correct data to file (tool, dateRented, fee) **Modular and maintainable (1 mark)** ♦ Modular and maintainable	**15**	Modular – four subprograms matching top level design. Maintainable must include – internal commentary, white space, meaningful variable/module names.

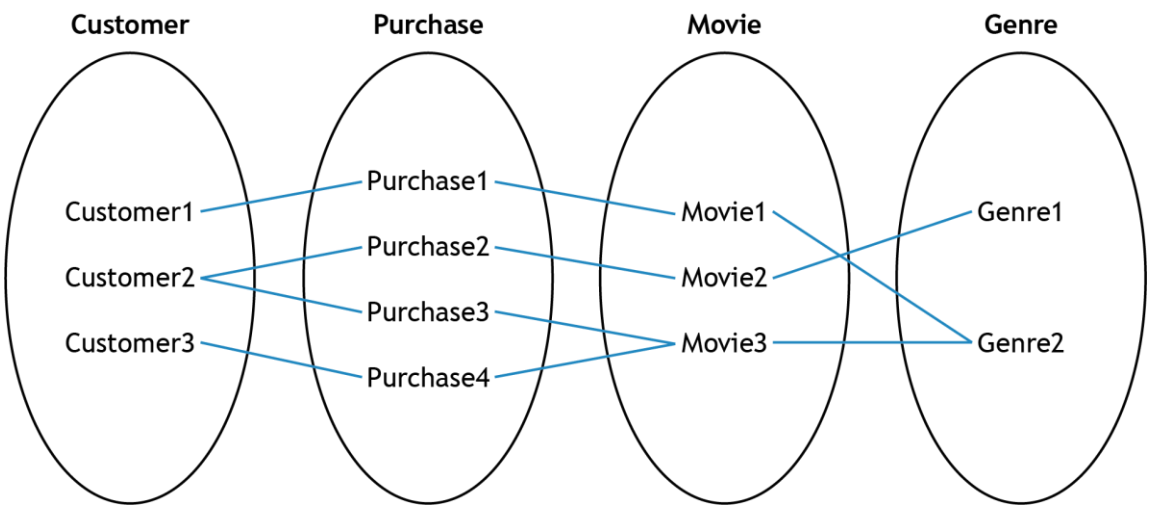
<table>
<tr><th>Task</th><th>Expected response</th><th>Max mark</th><th>Additional guidance</th></tr>
<tr><td>1di</td><td>See solution below.
<table>
<tr><th>index</th><th>If tool rented in 2025 or 2026 and not returned?</th><th>Current fee</th></tr>
<tr><td>Tool1</td><td>False</td><td>0</td></tr>
<tr><td>Tool2</td><td>True</td><td>25</td></tr>
<tr><td>Tool3</td><td>True</td><td>30</td></tr>
<tr><td>Tool4</td><td>True</td><td>25</td></tr>
</table></td><td>2</td><td>Award 1 mark for column two.
Award 1 mark for column three.</td></tr>
<tr><td>1dii</td><td>♦ (starting value set to £20 so) late fees for 2026 always have £20 added to them</td><td>1</td><td></td></tr>
<tr><td>1e</td><td>♦ (is not usable) as not displaying that the manufacturer is not found

OR

♦ (is usable) as displaying that the manufacturer is not found</td><td>1</td><td>Award mark if candidate describes solution to increase usability.

Answer must be consistent with candidate’s code.</td></tr>
</table>

Task 2 — database design and development

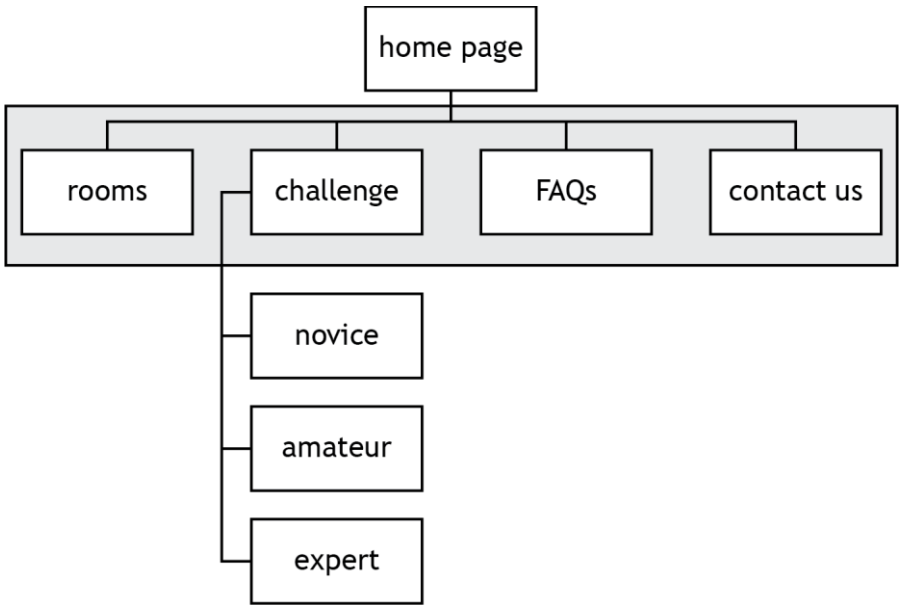
Task	Expected response	Max mark	Additional guidance
2a	**A query to:** ♦ find a genre with the word "suspense" in the description ♦ count the number of purchases ♦ calculate the average rating of each movie ♦ find the highest rated movie by a specific director ♦ search for premium accounts and calculate the total money spent on purchases ♦ find the shortest duration	2	Award **1 mark** for each bullet. Maximum**2 marks**.
2b	♦ adding the correct number of instances ♦ adding the correct associations 	2	

<table>
<tr><th>Task</th><th>Expected response</th><th>Max mark</th><th>Additional guidance</th></tr>
<tr><td>2c</td><td>♦ query to find the shortest duration
♦ fields, tables and equi-joins including query name
♦ search criteria (duration =) compared to first query alias/ sub query</td><td>3</td><td>For bullet two query name only needed in FROM clause if sub query has not been used.

Disregard any ORDER BY clause.</td></tr>
<tr><td></td><td colspan="3">Query2ci
<pre>SELECT MIN(duration) AS [Shortest Duration] FROM Movie;</pre>
Query2cii
<pre>SELECT forename, surname
FROM Customer, Purchase, Movie, Query2ci
WHERE Customer.customerID = Purchase.customerID AND
      Purchase.movieCode = Movie.movieCode AND
      duration = [Shortest Duration];</pre>
Subquery
<pre>SELECT forename, surname
FROM Customer, Purchase, Movie
WHERE Customer.customerID = Purchase.customerID AND
      Purchase.movieCode = Movie.movieCode AND
      duration = (SELECT MIN(duration) FROM Movie);</pre></td></tr>
<tr><td>2d</td><td>♦ fields, tables and ALIAS
♦ use of SUM function
♦ equi joins and search criteria
♦ GROUP BY all non-aggregate fields or unique identifier
♦ ORDER BY SUM(price) DESC</td><td>5</td><td>Don’t penalise if sorted by an incorrect aggregate function.

SQLite does allow ORDER By “Alias” or [Alias] but not ‘Alias’.</td></tr>
<tr><td></td><td colspan="3"><pre>SELECT forename, surname, SUM(price) AS [Total spent(£)]
FROM Customer, Purchase, Movie, Genre
WHERE Customer.customerID = Purchase.customerID AND
      Purchase.movieCode = Movie.movieCode AND
      Movie.genreID = Genre.genreID AND
      genreName = "Comedy"
GROUP BY forename, surname
ORDER BY SUM(price) DESC;</pre></td></tr>
<tr><td>2e</td><td>♦ correct range of release years
♦ grouping added using all non-aggregate fields or unique identifier</td><td>2</td><td>Range could be implemented as a range >=1990 and <=1999 or as a wildcard released like “199?” / “199_” / 199* / 199%</td></tr>
<tr><td></td><td colspan="3"><pre>SELECT forename, surname, email
FROM   Customer, Purchase, Movie
WHERE Customer.customerID = Purchase.customerID AND
       Purchase.movieCode = Movie.movieCode AND
       released like “199?”
GROUP BY forename, surname, email;</pre></td></tr>
<tr><td>2f</td><td>There is no director field.</td><td>1</td><td>Must reference field/attribute.</td></tr>
</table>

Task 3 — web design and development

Task	Expected response	Max mark	Additional guidance
3a	♦ home with link to Rooms, Challenge, FAQs and Contact us on level 1 ♦ novice, Amateur and Expert on level 2 of Challenge page ♦ indication of navigation bar  	3	
3b	♦ function(s)/in line JavaScript to show/hide image ♦ each overlay image hidden using onMouseOver event ♦ each overlay image shown using onMouseOut event	3	

	`function showJungleOverlay(){`
	`document.getElementById`
	`("jungleOverlay").style.display = "block";`
	`}`
	`function hideJungleOverlay(){`
	`document.getElementById`
	`("jungleOverlay").style.display = "none";}`
	
	`<section class="roomInfo" onmouseout="showJungleOverlay()">`
	
	`<img`
	`src="../Images/Overlay.png" class="overlay" id="jungleOverlay"`
	`onmouseover="hideJungleOverlay()"`
	`/>`		

Task	Expected response	Max mark	Additional guidance

3c	♦ float left and width 280px
	♦ 15px margin as per design
		2	Accept any solution that creates three columns side by side.
			
			Accept any solution that creates a 15px margin between the three sections.
			
			Possible solution
			`.challengeLevel {`
			`float: left;`
			`width: 280px;`
			`margin-left: 15px;`
			`background-color: #d4b483;`
			`}`

3di	♦ number of people ○ min="1" ○ max="5" ♦ select for rooms ○ Multiple ♦ radio buttons challenge ♦ correct positions and required applied to the three fields (number of people, rooms and challenge)	**4**	Validation for each added form element is necessary to achieve the mark. Radio button could have one option set to checked to ensure required.
3dii	**Select (dropdown menu) checked for:** ♦ single selection ♦ multiple selection ♦ left blank	**1**	
3e	♦ the expert link on the 'Challenge' page does not link to its respective page ♦ the 'Amateur' page does not contain customer quotes ♦ footer is not consistent across all pages ('Contact us' page footer is missing)	**2**	Award **1 mark** for each bullet. Maximum**2 marks**. Accept answers that relate to the candidate's own code.

[END OF MARKING INSTRUCTIONS]