Higher
Coursework
Assessment Task

Higher Computing Science
Assignment
Finalised Marking instructions

Marking instructions

In line with SQA's normal practice, the following marking instructions for the Higher Computing Science assignment are addressed to the marker. They will also be helpful for those preparing candidates for course assessment.

Candidates' evidence is submitted to SQA for external marking.

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.

`

Specific marking instructions

Task 1 — software design and development

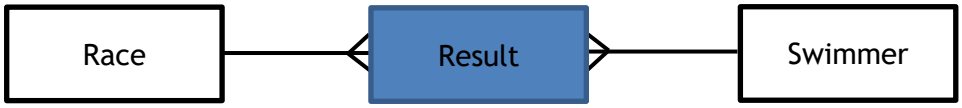
Task	Expected response	Max mark	Additional guidance
1a	♦ input of name of chosen company AND output of highest number of employees ♦ calculate the difference between the highest paid CEO and the searched CEO	**2**	
1b	♦ initialise position variable ♦ loop and conditional statement to find max ♦ update variable ♦ return position of maximum	**4**	Ignore additional design.

Task	Expected response	Max mark	Additional guidance
1c	Read in company data **(2 marks)** ♦ module with parameters passed or returned ♦ assigned to three parallel arrays Find maximum position function **(3 marks)** ♦ module with single formal parameter ♦ loop to traverse to end of array with if statement to find correct maximum position ♦ maximum position returned Calculate salary difference **(4 marks)** ♦ module with correct parameters passed in (company, ceoSalary), nothing returned (message displayed within procedure) ♦ array searched for selected company position ♦ use of max position and found position to calculate difference between selected company salary and max salary ♦ use of flag variable to implement not found message Find the highest number of employees **(3 marks)** ♦ module with correct parameter passed in (numEmployees) and nothing passed out (message displayed within procedure) ♦ count initialised and incremented ♦ use of max position in condition to find those within 10% Implementation **(3 marks)** ♦ single find max function called twice ♦ function(s) called from within each subprocedure ♦ modular and maintainable	**15**	Award 0 marks for bullet 1 if global variables are used in function. If maximum value is returned do not penalise its use in • salary difference bullet 3 Accept use of max value/chosen value in difference calculation if maxpos/chosen pos has been used to assign value. If maximum value is returned do not penalise its use in • highest number of employees bullet 3 Accept use of max num employess in calculation if maxpos has been used to assign value. Maintainable includes internal commentary, white space, meaningful variable and procedure names.

Task	Expected response	Max mark	Additional guidance
1d(i)	♦ two companies have the same highest salary, only one would be displayed/ doesn't allow for the possibility of multiple CEO's salary being the same as the maximum **OR** ♦ the function only returns a single maxPos (rather than an array of positions)	**1**	Do not accept two companies have the same salary.
1d(ii)	♦ traverse ceoSalary array with a loop ♦ compare ceoSalary against maximum salary **OR** array of max positions used	**2**	Indication of salary must be in either bullet one or two.
1e	♦ finding max algorithm is written once but called twice ♦ local variables discarded after function is executed ♦ is inefficient as the code has been repeated twice	**1**	Award 1 mark any one bullet. Answers must be relevant to candidate's code.

Task 2 — database design and development

<table>
<tr><th>Task</th><th>Expected response</th><th>Max mark</th><th>Additional guidance</th></tr>
<tr><td>2a</td><td>A query to:
♦ sort results of a race by position
♦ calculate time differences between swimmers' time and the winning time
♦ search for swimmer team reference number
♦ search for first, second and third placed swimmers
♦ calculate the total number of days a city has hosted the event
♦ update existing data on events, races and swimmers</td><td>2</td><td>Award 1 mark for each bullet. Maximum 2 marks.

Functional requirements should be extracted from end user information.</td></tr>
<tr><td>2b(i)</td><td>Race to Result (1:M) and
Swimmer to Result (1:M)
 
</td><td>1</td><td></td></tr>
<tr><td>2b(ii)</td><td>raceNumber and swimmerID</td><td>1</td><td></td></tr>
<tr><td>2c</td><td>♦ fields, tables and ALIAS
♦ COUNT function
♦ equi joins and search criteria
♦ GROUP BY all non aggregate fields or unique identifier

<pre>SELECT initial, surname, swimCategroy, teamName,
       COUNT(*) AS [Races won]
FROM   Result, Swimmer, Team
WHERE Result.swimmerID = Swimmer.swimmerID AND
        Swimmer.teamRef = Team.teamRef AND
        position = 1
GROUP BY initial, surname, swimCategory, teamName</pre>
OR
<pre>GROUP BY Swimmer.swimmerID</pre></td><td>4</td><td>Award bullet 2 if SUM(position) used as will work due to position being 1.</td></tr>
</table>

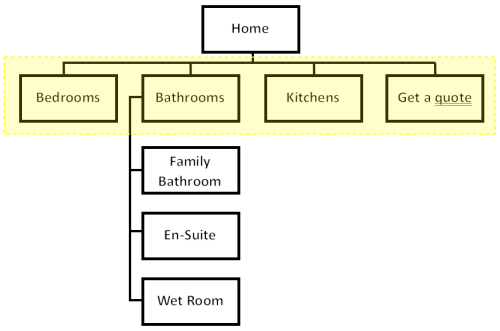
Task	Expected response	Max mark	Additional guidance
2d	♦ query to find correct fastest time (0:22.79) ♦ fields,tables and equi joins including query name/sub query ♦ search criteria (lane = 1 or lane = 8) in both queries ♦ use of first Query as alias in WHERE clause OR sub query	**4**	Query using MIN could be a sub-query Award 0 marks for bullet 4 if value 0:22.79 is used instead of field

```
SELECT MIN(raceTime) AS fastestTime
FROM   Result
WHERE  (lane = 1) OR (lane = 8);

SELECT initial, surname, teamName, city, eventDate
FROM   Event, Race, Result, Swimmer, Team, [2 e i]
WHERE  Event.eventID = Race.eventID AND
       Race.raceNumber = Result.raceNumber AND
       Result.swimmerID = Swimmer.swimmerID AND
       Swimmer.teamRef = Team.teamRef AND
       time = fastestTime AND (lane = 1) OR (lane = 8);
```

2e	♦ search criteria added to the where clause ♦ ORDER BY COUNT()/ALIAS	**2**	Multiple ways to express position being 1,2 or 3
2f	♦ add end date/number of days field to the Event table **OR** ♦ add a season field to the Event table	**1**	Must mention Event table

Task 3 — web design and development

Task	Expected response	Max mark	Additional guidance
3a	♦ home with links to Bedrooms, Bathrooms, Kitchens and Get a quote on level 1 ♦ family Bathroom, En-Suite and Wet Room on level 2 of Bathrooms page ♦ indication of navigation bar	**3**	
3b	♦ bedroom descriptions hidden when page loads ♦ classes / IDs added to all 3 bedrooms ♦ sections revealed using onclick event on each image element ♦ three functions to display corresponding description or a single function with appropriate parameters	**4**	Ignore additional code to hide other descriptions
3c	♦ padding/margin (10 pixels) ♦ width and float applied to position text ♦ text-align center applied to text	**3**	Do not accept use of margin/padding to replace width property
3d	♦ required added to four fields ♦ multiple added to rooms ♦ name, email address maxlength set to 40 chars, description changed to 19 rows	**3**	
3e	♦ wet room link not working ♦ no reviews ♦ before/after image error on kitchen page	**2**	Award 1 mark for each bullet. Maximum 2 marks.

[END OF MARKING INSTRUCTIONS]

Candidate: **Total marks awarded:** **/40**

Task 1 – Software Design and Development

		Marks Available	Marks Awarded	
1a – Analysis	Input of chosen company name AND output of highest number of employees	1		
	Calculate difference between salaries	1		**/2**

1b – Design	Initialise position variable	1		
	Loop and conditional statement	1		
	Update of variable	1		
	Return position of maximum	1		**/4**

1c – Implementation:

Read in company data	Module with parameters	1		
	Assigned to three parallel arrays	1		
Find max function	Module with single formal parameter	1		
	loop to traverse entire array with if statement to find maximum position	1		
	Return maximum position	1		
Calculate salary difference	Module with two parameters	1		
	Linear search to find position of company	1		
	Max position and found position to calculate difference	1		
	Flag variable to display not found	1		
Find highest number of employees	Module with one parameter	1		
	Count initialised and incremented	1		
	Max position in condition to find 10%	1		
Implementation	FindMaxPos function called twice	1		
	Function called from within each subprocedure	1		
	Modular and maintainable	1		**/15**

1d – Testing (i)	Refinement only one/not multiple or Function only returns a single position	1		**/1**
1d – Testing (ii)	Traverse salaries array with a loop	1		
	Compare ceoSalary against maximum salary or array of max positions used	1		**/2**

1e – Evaluation	Efficiency	1		**/1**

Task 2 – Database Design and Development

		Marks Available	Marks Awarded	
2a – Analysis	Functional requirement 1	1		
	Functional requirement 2	1		/2
2b(i) – Design	Cardinality and relationships	1		/1
2b(ii) – Design	raceNumber and swimmerID	1		/1
2c – Implementation	Fields, tables and alias	1		
	COUNT function	1		
	Equi joins and search criteria	1		
	GROUP BY all fields/unique identifier	1		/4
2d – Implementation	Query to find fastest time (0:22.79)	1		
	Fields, tables and equi joins	1		
	Search criteria in both queries (lane = 1 or lane = 8)	1		
	Use of first query	1		/4
2e – Testing	Search criteria added to WHERE clause	1		
	ORDER BY COUNT()/ALIAS	1		/2
2f – Evaluation	Additional field required in Event table	1		/1

Task 3 – Web Design and Development

		Marks Available	Marks Awarded	
3a – Design	Home with links to Bedrooms, Bathrooms, Kitchens and Get a quote on level 1	1		
	Family Bathroom, En-Suite and Wet Room on level 2 of Bathrooms page	1		
	Indication of Nav Bar	1		/3
3b – Implementation	All bedroom text hidden	1		
	Classes/IDs added	1		
	Onclick event	1		
	Function(s) to display	1		/4
3c – Implementation	Padding/margin	1		
	Width, float applied	1		
	Text-align: center added to text	1		/3
3d – Testing	Required added to all fields	1		
	Multiple added to rooms	1		
	name, email address maxlength set to 40 chars, description changed to 19 rows	1		/3
3e – Evaluation	Not fit for purpose reason	**2**		/2