



Higher  
Coursework  
Assessment Task



# 2019 Higher Computing Science Assignment Finalised Marking Instructions

© Scottish Qualifications Authority 2019

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

The information in this document may be reproduced in support of SQA qualifications only on a non-commercial basis. If it is reproduced, SQA must be clearly acknowledged as the source. If it is to be reproduced for any other purpose, written permission must be obtained from [permissions@sqa.org.uk](mailto:permissions@sqa.org.uk).



# Marking instructions

Marking instructions are provided for this specimen assessment task. In line with SQA's normal practice, they are addressed to the marker. They will also be helpful for those preparing candidates for course assessment.

Marking instructions will not be provided with annual assessment tasks, as candidate evidence will be submitted to SQA for external marking.

## General marking principles

This information is provided to help you understand the general principles that must be applied when marking candidate responses in this assignment. These principles must be read in conjunction with the specific 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 specific marking instructions for this assessment.
- b Marking should always be positive. This means that, for each candidate response, marks are accumulated for the demonstration of relevant skills, knowledge and understanding: they are not deducted from a maximum on the basis of errors or omissions.

# Specific marking instructions

## Part 1 – Database design and development

Task	Expected response	Additional guidance	Max Marks
1. (a)	1 mark for completing entity names, in the correct order adding the correct number of instances adding the correct associations	Note That the entities could be written in reverse order (Route-Flight-Booking-Customer).	3

Design (3)

Task	Expected response	Additional guidance	Max Marks	Implementation (7)
1. (b) (i)	<p>1 mark for each bullet</p> <ul style="list-style-type: none"> <li>• Three fields displayed forename surname Tax</li> <li>• Calculation correctly implemented using field names</li> <li>• Equi-join(s) and criteria produce correct output in single row</li> </ul>	<p>SELECT forename, surname, (adultTicket*5.50)+(childTicket*2.0)+(concessionTicket*1.5) AS Tax (£ not required) FROM Customer, Booking, Flight WHERE Customer.customerID = Booking.customerID AND Booking.flightID = Flight.flightID AND Customer.customerID = "GR01932" AND Flight.flightID = "QH182";</p> <p>Note the following:</p> <ul style="list-style-type: none"> <li>• Fields can come from different tables using dot notation. Equi-joins may vary.</li> <li>• Access Query Design, award first two marks</li> </ul>	3	
(ii)	<p>Award 1 mark for each bullet:</p> <ul style="list-style-type: none"> <li>• Query to find the maximum number of children</li> <li>• Correct fields and tables</li> <li>• Equijoin</li> <li>• Use of Query1 result in the criteria to produce the correct output</li> </ul>	<p>Query using MAX function could be a sub-query within the criteria</p> <p>SELECT Max(childTicket) AS maxChildren FROM Booking;</p> <p>SELECT Customer.forename, Customer.surname FROM Customer, Booking, (Query1)</p> <p>WHERE Customer.customerID= Booking.customerID</p> <p>AND Booking.childTicket = Query1.alias</p> <p>Do not award a mark for the final bullet point if the candidate answers "AND childTicket = 8".</p> <p>Access Query Design, only award first mark</p>	4	

Task	Expected response	Additional guidance	Max Marks	Evaluation (2)
1. (c)	<p>1 mark each for any realistic answer, for example:</p> <ul style="list-style-type: none"> <li>• there would be no limit on the number of tickets that could be booked on one flight ie capacity exceeded</li> <li>• arrivalDate/Time before departureDate/Time</li> <li>• format of date and time incorrect (eg American format)</li> <li>• invalid/no data assigned to fields</li> <li>• invalid RouteID could be entered if there is no link to Route table (referential integrity)</li> </ul>	Answer must refer to fields in Flight table	2	

Part 2 Software design and development

Task	Expected response	Additional guidance	Max Marks	
2. (a)	<p>One mark for each complete bullet point:</p> <p><b>Inputs</b></p> <ul style="list-style-type: none"> <li>• forename, surname and distance/total miles from file</li> </ul> <p><b>Processes</b></p> <ul style="list-style-type: none"> <li>• find the furthest distance walked AND the members that walked further than 70% of furthest distance</li> </ul> <p><b>Outputs</b></p> <ul style="list-style-type: none"> <li>• (display) the furthest distance walked AND write the forename, surname (or names) of those who walked further than 70% of furthest distance (to file)</li> </ul>	<p>Ignore additional processes</p> <p>Do not accept members</p>	3	Analysis (3)
(b)	<p>1 mark each for completing the following.</p> <ul style="list-style-type: none"> <li>• members record passed into "Find the furthest distance walked"</li> <li>• furthest distance passed into "Write club prize winners to file"</li> </ul>	<p>Accept members without fields specified or members (distance) but not distance without members</p> <ul style="list-style-type: none"> <li>• Two correct parameters, no errors - 2 marks</li> <li>• Two correct parameters, one incorrect, award 1 mark</li> <li>• One correct parameter only, award 1 mark</li> <li>• One correct parameter, one incorrect, award 1 mark</li> <li>• Anything else = 0 marks</li> </ul>	2	

Sample Answer		
Read members data from file into array of records	IN	
	OUT	members(forename,surname,distance)
Find the furthest distance walked	IN	members(forename,surname,distance)
	OUT	furthest
Display the furthest distance walked	IN	furthest
	OUT	
Write club prize winners to file (Linear search)	IN	members(forename,surname,distance) furthest
	OUT	

Task	Expected response		Additional guidance	Max Marks
2. (c) (i)	Input from file to array of records	Module with correct parameter passed or returned to read data from file to array of records		1
		Correctly assign data from file to array of records		1
		Member data stored in a record structure		1
	Find Furthest Distance	Module with correct parameter passed to return furthest distance walked		1
		Find max algorithm.  Award 3 marks for: <ul style="list-style-type: none"> <li>• Initialise furthest</li> <li>• Correct assignment of furthest within loop</li> <li>• Matches design - initialise to first index in array - loop from second index in array</li> </ul>		3
	Display Furthest	Module with correct parameter passed to display furthest distance walked		1
	Linear Search with File Output	Module with correct parameters		1
		Award 2 marks for:  selection - record distance more than 70% of furthest walk  write forename(s) and surname(s) to file		2
	Program Code is Maintainable (must have meaningful variable names and regular internal commentary)			1
	Matches top level design: four sub programs with furthest distance as a function			4

Implementation (13)

Task	Expected response		Additional guidance	Max Marks	Implementation (2)
2. (c) (ii)	Marathon Task	Correct use of function to truncate	Round function does not truncate	1	
		Message, forename(s), surname(s) and number of whole marathons written to file		1	
<b>Steven Johnston should be 0</b>					
(d)	One mark each for: <ul style="list-style-type: none"> <li>Using variable names for furthest distance and member's record structure from candidate's code AND first value of max distance and furthest assigned to 189.4</li> <li>Table shows correct changing state of member distance and furthest</li> </ul>		Where value of furthest does not change, it can be omitted from the trace	2	Testing (2)
(e)	Fitness for Purpose  Award 1 mark any two bullets from: <ul style="list-style-type: none"> <li>Find/display furthest distance walked</li> <li>Find/display/write members who walked more than 70% of the furthest distance</li> <li>Read forename, surname, distance from file</li> <li>Find number of marathons OR write names and number of marathons to file</li> </ul> Award 1 mark linking readability to maintainability eg internal commentary to explain code to another programmer  Award 1 mark linking modularity to maintainability eg sub procedures used (can be edited independently)		Comments on not fit for purpose should match functional requirements           Must go beyond a list of readability criteria	3	



Part 3 Web design and development

Task	Expected response	Additional guidance	Max Marks	
3. (a)	<p>Functional requirements could include two of the following for 1 mark each:</p> <ul style="list-style-type: none"> <li>the website should display text about the work of the foundation</li> <li>the website should display text on the history of playing cards</li> <li>the website should display a variety of appropriate graphics</li> <li>the website should display text detailing the rules of several different card games</li> <li>the website should collect users contact information (using a form)</li> <li>the website should show and hide objects in response to mouse clicks</li> </ul>	Must have some element of context eg not just pictures but pictures of playing cards	2	Analysis (2)
(b) (i)	<p>Confirm the following for 1 mark each:</p> <ul style="list-style-type: none"> <li>All three sections correctly positioned</li> <li>10 pixel margin between Early History and European Adoption</li> <li>10 pixel margin between Early History/European Adoption and Modern Cards</li> <li>germanPlayingCards graphic floated right and level with Modern Cards</li> </ul>	<p>Positioning of sections could be done by:</p> <ul style="list-style-type: none"> <li>Early History float left</li> <li>European Adoption float right</li> <li>Early History float left, increase margin on European Adpotion</li> <li>Changing the width of sections and set display to inline</li> </ul> <p>10 pixels can be implemented in several ways.</p>	4	
(ii)	<p>Award 1 mark each for:</p> <ul style="list-style-type: none"> <li>IDs assigned to sections</li> <li>three sections set to display:none</li> <li>onclick events added to images</li> <li>three functions each displaying one element while hiding the others OR a single function with appropriate parameters</li> </ul>		4	

Task	Expected response	Additional guidance	Max Marks	
3. (c)	Award 1 mark for any 3 bullets: <ul style="list-style-type: none"> <li>• the select element should be tested to ensure that multiple card games can be selected:</li> <li>• the maximum length of the text inputs should be tested using valid/invalid data</li> <li>• the range of acceptable values in the numeric input (years) should be tested with valid/invalid data</li> <li>• the required inputs should be tested with null values</li> </ul>	No test on age  For years test, must include at least two types of data (in range and out of range)	3	

Task 1 - Database Design and Development		Marks Available	Marks Awarded
1a - design	Entity	1	
	Instances	1	
	Associations	1	
<b>1a - design total</b>		<b>3</b>	
1b(i) - implementation	Three fields including alias Tax	1	
	Calculation	1	
	Equijoin and unique criteria	1	
<b>1b(i) - implementation total</b>		<b>3</b>	
1b(ii) - implementation	Query to find max children	1	
	Fields and tables second query	1	
	Equijoin	1	
	Criteria for correct output	1	
<b>1b(ii) - implementation total</b>		<b>4</b>	
<b>1c - evaluation</b>		<b>2</b>	
Task 2 - Software Design and Development		Marks Available	Marks Awarded
2a - analysis	Inputs	1	
	Processes	1	
	Outputs	1	
<b>2a - analysis total</b>		<b>3</b>	
2b - design	Data Flow	2	
<b>2b - design total</b>		<b>2</b>	
<b>2ci - implementation</b>			
Input from file	Module with correct parameter	1	
	Assign data from file	1	
	Member data stored in record structure	1	
Find furthest distance	Module with correct parameter	1	
	Initialise furthest	1	
	Correct assignment of furthest	1	
	Matches design	1	
Display furthest	Module with correct parameter	1	
Linear search with file output	Module with correct parameters	1	
	Selection	1	
	Write correct names to file	1	
Program code is maintainable		1	
Matches top level design	4 sub programs (1 function)	1	
<b>2c(i) - Implementation total</b>		<b>13</b>	
2c(ii) - implementation	Pre-defined function	1	
	Message and names written to file	1	
<b>2c(ii) - Implementation total</b>		<b>2</b>	

2d - testing	Suitable column headings and initial values	1	
	Distance and Furthest traced	1	
<b>2d - testing total</b>		<b>2</b>	
2e - evaluation	Fitness for purpose	1	
	Readability and Modularity	2	
<b>2e - evaluation total</b>		<b>3</b>	

<b>Task 3 - Web Design and Development</b>		<b>Marks Available</b>	<b>Marks Awarded</b>
3a - analysis	Functional requirements	2	
<b>3a - analysis total</b>		<b>2</b>	
3b(i) - implementation	Three sections correctly positioned	1	
	10 pixel margin between Early History and European Adoption	1	
	10 pixel margin between Early History/European Adoption and Modern Cards	1	
	Image floated right	1	
<b>3b(i) - implementation total</b>		<b>4</b>	
3b(ii) - implementation	IDs assigned	1	
	Three sections set to None	1	
	Onclick events	1	
	Procedure shows one element hiding the others	1	
<b>3b(ii) - implementation total</b>		<b>4</b>	
3c - testing	Three different types of tests	3	
<b>3c - testing total</b>		<b>3</b>	

	<b>Marks Available</b>	<b>Marks Awarded</b>
<b>Assignment total</b>	<b>50</b>	

[END OF MARKING INSTRUCTIONS]