Higher
Coursework
Assessment Task

2020 Higher Computing Science Assignment Marking Instructions

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

Please note, as we were not able to carry out live marking in 2020, these marking instructions are not presented in a final state and have not been referenced against candidate responses.

Marking instructions

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

Task	Expected response	Additional guidance	Max marks	
1ai	Award 1 mark for all Car attributes: ♦ registration number ♦ make of car ♦ model of car ♦ year manufactured	Award 0 marks if attributes from other entities are included.	1	Analysis (3)
1aii	Award 1 mark each for any two bullets: ♦ A query to sort cars by year of manufacture ♦ A query to count the number of days a car is in the garage (using an aggregate function) ♦ Update/insert customer details ♦ Search for details of garages/ customers/cars/jobs	Functional requirements must be relevant to end-user requirements specified.	2	

Task	Expected response	Additional guidance	Max marks	

1bi	Award 1 mark for each bullet:
	
	♦ Fields and alias
	- garageName
	- Total sales
	♦ SUM function using cost field
	♦ Equi-join and date criteria
	♦ Group by garageName
		`SELECT garageName, SUM(cost) AS [Total sales] FROM Job, Garage WHERE dateOut Like "19/01/2020" AND Job.garageID = Garage.garageID GROUP BY garageName;`
		
		Accept alterntiave date formats e.g.
		`dateOut = #1/19/2020#`
		`dateOut Like "19/01/2020*"`
		`dateOut = "20/01/19"`
			4	Implementation (8)

1bii	Award 1 mark for each bullet:
	
	♦ Query to find number of days using Max function
	♦ Correct fields and tables
	♦ Equi-join
	♦ Use of Query 1 result in the criteria to produce correct output
		`SELECT Max(dateOut-dateIn) AS [Number of Days] FROM Job;`
		
		`SELECT [Number of Days], Job.regNo, garageName FROM Query 1, Job, Garage WHERE (dateOut-dateIn) = [Number of Days] AND Job.garageID = Garage.garageID;`
		
		Accept alternative date calculations e.g.
		`DATEDIFF(dateOut, dateIn)`
		`DATEDIFF(day,dateOut,dateIn)`
			4	

1ci	Query does not group by unique identifier **OR** Query groups by customer forename and surname but multiple customers have the same name		1	Testing (2)
1cii	**Customer.customerID** should be added to the GROUP BY clause		1	

Task	Expected response	Additional guidance	Max marks	
2a	Award 1 mark each for any two bullets: ♦ Member data from file is accurate so there is no need to validate this data ♦ Maximum membership is 50 ♦ Only one member is added each time the program runs ♦ New member details are not written back to file ♦ Only three types of category are valid		2	**Analysis (2)**
2b	Award 1 mark for each bullet: ♦ forename,surname, category, password passed out of first procedure ♦ forename,surname, category, password passed in to second procedure ♦ category() passed out of second procedure	Award 3 marks only when no additional or incorrect parameters are given. If all three bullets are correct but additional, incorrect parameter(s) are added award 2 marks. New member data not arrays Final bullet must indicate an array	3	**Design (3)**

Task	Expected response	Additional guidance	Max marks	
2c	Call validate password function **from within** first procedure		1	Implementation (15)
	Extract/identify first and last character	Accept use of pre-defined function or array position	1	
	Use of pre-defined function to convert first and last characters to ASCII		1	
	Validate first and last characters		1	
	Return password		1	
	Parameters passed to second procedure		1	
	♦ Read data from file ♦ Assign to parallel arrays		2	
	Add/append new member data to existing arrays		1	
	Display all members, count of category and total members	'Oliver Wilson Adult' showing as new member Award 0 marks if password is displayed	1	
	Parameter category() passed to third procedure		1	
	Counting occurrence algorithm: ♦ Intialise three count variables ♦ Increment appropriate count		2	
	Program code is maintainable	Meaningful variable names and regular internal commentary should be relevant to task.	1	
	Matches steps 1, 2 and 3 of main algorithm		1	

Task	Expected response	Additional guidance	Max marks	
2d	Award 1 mark each for any two bullets: Design tests to ensure that: ♦ the correct substrings are extracted/first and last characters correctly identified ♦ characters are correctly converted to ASCII code ♦ the first and last characters are each within the valid ASCII value range ♦ both first and last characters have to be valid to make the whole password valid ♦ the valid password is returned to main procedure	Answers should reference/give examples from candidate's code.	2	**Testing (2)**
2e	Award 1 mark for an example from each category. This must be evidenced in candidate's code. ♦ Efficiency eg: - use of appropriate constructs (loops, else if statements) - use of appropriate data structures (parallel arrays) - parameter passing (reusable modules, no global variables) ♦ Robustness eg: - input validation ♦ Fit for purpose This must refer back to functional requirements. Program must run as expected	Accept not efficient/robust/fit for purpose responses with valid reasons/examples from code	3	**Evaluatoin (3)**

Task	Expected response	Additional guidance	Max marks	
3a	Award 1 mark for each bullet: ♦ Text and image added to wireframe ♦ List to **the left** of each image		2	**Design (2)**
3bi	Award 1 mark for each bullet: ♦ Hide text elements on loading ♦ Onmouseover/onmouseout events to reveal/hide text		2	**Implementation (7)**
3bii	Award 1 mark for each bullet: ♦ Drop down menu with six options ♦ Multiple selection of toppings ♦ Range check on quantity ♦ Radio buttons for delivery/collection ♦ Submit button with 'Add to basket' value		5	

Task	Expected response	Additional guidance	Max marks	
3c	Award 1 mark for any two bullets: ♦ Check that three options are displayed when form loads ♦ Check that options are displayed as a drop-down menu ♦ Check that multiple toppings can be selected		1	**Testing(1)**
3d	Award 1 mark each for any two bullets: Home page ♦ Links to download apps do not work Our Pizza page ♦ No information on sourcing ingredients Reviews page ♦ No links ♦ Not able to add review ♦ Not able to search for review		2	**Evaluation (2)**

[END OF MARKING INSTRUCTIONS]