Higher
Coursework
Assessment Task

Higher Computing Science
Assignment
Finalised Marking instructions

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.

Specific marking instructions

Task 1 — software design and development

<table>
<tr><th>Task</th><th>Expected response</th><th>Max mark</th><th>Additional guidance</th></tr>
<tr><td>1a</td><td>Input
♦ month to be searched
Process
♦ find the first 5 star review for the given month
♦ count the total number of deliveries and total number of collections</td><td>3</td><td>Both counts needed for the third bullet.</td></tr>
<tr><td>1b</td><td>♦ Step 1 and Step 2 (1 mark)
<table>
<tr><td>1</td><td>IN</td><td></td></tr>
<tr><td></td><td>OUT</td><td>Orders(...)</td></tr>
<tr><td>2</td><td>IN</td><td>Orders(...)</td></tr>
<tr><td></td><td>OUT</td><td>position</td></tr>
</table>
♦ Step 3 and Step 4 (1 mark)
<table>
<tr><td>3</td><td>IN</td><td>Orders(...), position</td></tr>
<tr><td></td><td>OUT</td><td></td></tr>
<tr><td>4</td><td>IN</td><td>Orders(...)</td></tr>
<tr><td></td><td>OUT</td><td></td></tr>
</table></td><td>2</td><td>Accept orders() or orders(orderNum, date, email, option, cost, rating) or orders(date, rating)

Do not accept () after position in step 3.

Award 0 for either bullet point if there is any additional data flow.</td></tr>
</table>

Task	Expected response	Max mark	Additional guidance
1c	Read Data **(2 marks)** ♦ Module with parameter passed or returned ♦ Assigned to array of records Find First 5 Star Review **(5 marks)** ♦ Function with correct parameter (orders()) ♦ Position and index initialised and updated ♦ Conditional loop with two correct conditions ♦ Comparison of entered month with month text from data structure and comparison of currentRating = 5 ♦ Position returned Write to File **(3 marks)** ♦ Module with correct parameters (orders(), position) ♦ Open and close new file ♦ Write correct details (orderNum, email, cost) of winning customer or 'no winner' to file Count and display totals **(4 marks)** ♦ Module with correct parameter (orders()) and display of both totals ♦ Single count function called twice ♦ Count function with correct parameters (orders, collection/delivery) and count returned ♦ Count occurrence for collection and delivery Modular and maintainable **(1 mark)** ♦ Modular and maintainable	**15**	position = - 1 and index < length array Delivery – 290, collection - 215 Do not award mark for use of predefined function instead of count occurrence algorithm Modular – modules must be called in the correct order and without use of global variables 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>1d</td><td>

rows 2 and 3
row 4

<table>
<tr><th>orderNum</th><th>position</th><th>IF…</th></tr>
<tr><td>Ord165</td><td>-1</td><td>False</td></tr>
<tr><td>Ord166</td><td>-1</td><td>False</td></tr>
<tr><td>Ord167</td><td>-1</td><td>False</td></tr>
<tr><td>Ord168</td><td>3</td><td>True</td></tr>
</table>
</td><td>2</td><td></td></tr>
<tr><td>1e</td><td>
Efficiency (1 mark)
Award one mark for any correct bullet.

My program is efficient because:

one function is called twice
the extra parameter allows it to count both options

OR
My program is inefficient because:

I programmed two separate functions (instead of using one)
I could have counted both in one iteration of the array

Maintainability
Data structure (1 mark)

array size is fixed (505 elements) and would need to be updated

OR

the array is initialised with no size and new elements are appended/arrays are redimensioned

Loop (1 mark)

number of iterations of loops are fixed (505)

OR

the loop iterates to the length of the array/end of file

</td><td>3</td><td>Answer must correspond to candidate’s own code.

Accept reverse of any bullet.

Accept explanation for single iteration through array to find both totals (can be as an expression of inefficiency or efficiency).</td></tr>
</table>

Task 2 — database design and development

Task	Expected response	Max mark	Additional guidance
2a	A query to: ♦ calculate average value of the comics (in the collection) ♦ calculate the total value of comics with a specific (main) character	**2**	
2b	To avoid a many to many relationship (between comic and characterInfo)	**1**	
2c	♦ query to calculate average ♦ fields, tables, equi-join and use of query name/sub-query ♦ valuation >= calculated average + 300	**3**	Candidates may +300 at first bullet. Candidates may also include `ORDER BY comicTitle`

Query1

```
SELECT AVG(valuation) AS avgValuation FROM Comic;

SELECT comicTitle, issue, publisherName, valuation
FROM Comic, Publisher, Query1
WHERE valuation >= avgValuation + 300
AND Comic.publisherID = Publisher.publisherID
```

Sub Query

```
SELECT comicTitle, issue, publisherName, valuation
FROM Comic, Publisher
WHERE valuation >= (SELECT AVG(valuation) FROM Comic) + 300 AND
Comic.publisherID = Publisher.publisherID
```

Task	Expected response	Max mark	Additional guidance
2d	♦ use of SUM with alias ♦ fields, tables and joins and main character criteria ♦ wildcard ♦ group by ♦ sort on calculation	5	Accept `mainCharacter = Yes/True` If main character criteria has been missed, output will include 'Iron Duck'. Allow Access wildcard *Duck*. Allow `GROUP BY characterID.`

	`SELECT characterName,SUM(valuation) AS 'Total Valuation'`
	`FROM CharacterInfo,Comic,ComicCharacter`
	`WHERE characterName LIKE '%Duck%' AND mainCharacter = 1 AND`
	`CharacterInfo.characterID = ComicCharacter.characterID AND`
	`ComicCharacter.comicID = Comic.comicID`
	`GROUP BY characterName`
	`ORDER BY SUM(valuation) DESC;`		

2e	♦ missing equi join `ComicCharacter.comicID = Comic.comicID` ♦ `valuation * 2`	2	Allow `SUM(valuation)*2` with `GROUP BY`
2f(i)	A query to calculate how much a comic has increased in value since it was purchased	1	
2f(ii)	A field added to the Comic entity to store the price/value when purchased	1	Must reference comic table

Task 3 — web design and development

Task	Expected response	Max mark	Additional guidance
3a	♦ javaScript to reveal relevant record score ♦ form to submit an enquiry	**2**	Accept named JavaScript events.
3b	♦ 3 Sections side by side ♦ classes/ID's added ♦ 10 pixel margins between sections ♦ matches Design (Width 31%, height 300px, background colour #4ABBD9, padding 5px)	**4**	See prices.html and style.css files. Can be styled with descendent selectors.
3c	♦ function(s)/in line JavaScript to change image ♦ function(s)/in line JavaScript to reset image to scorecard ♦ mouseOver event to change, mouseOut event to reset	**3**	See scorecard.html.
3d	Contact number ♦ cannot exceed max length/11 characters ♦ cannot be left blank Choose your enquiry ♦ both one and multiple items can be selected ♦ cannot be left blank	**3**	Award 1 mark for each bullet. Maximum 3 marks. Reference to none, one and multiple selections can be awarded third and fourth bullets.
3e	♦ cannot opt-out of mailing list ♦ no working link to Prices page (links to home page) ♦ 'View Menu' button doesn't show menu ♦ no option for Kids menu on food page ♦ cannot submit enquiry form ♦ no pictures of facilities	**3**	Award 1 mark for each bullet. Maximum 3 marks.

[END OF MARKING INSTRUCTIONS]