



National
Qualifications
SPECIMEN ONLY

S816/77/11

Computing Science

Duration — 2 hours

Total marks — 55

SECTION 1 — Software design and development — 35 marks

Attempt ALL questions.

Attempt EITHER Section 2 OR Section 3

SECTION 2 — Database design and development — 20 marks

SECTION 3 — Web design and development — 20 marks

You may use a calculator.

Write your answers clearly in the answer booklet provided. In the answer booklet you must clearly identify the question number you are attempting.

Use **blue** or **black** ink.

Before leaving the examination room you must give your answer booklet to the Invigilator; if you do not, you may lose all the marks for this paper.



* S 8 1 6 7 7 1 1 *

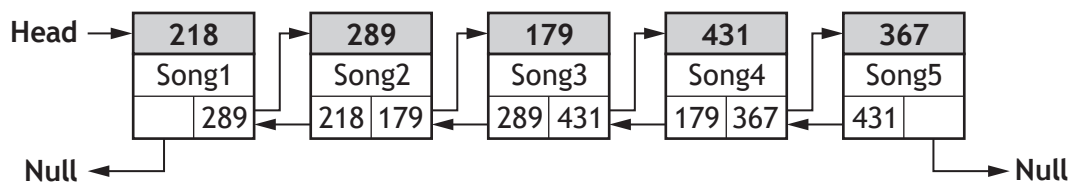
Attempt ALL questions

1. A DJ app allows users to create a playlist. The playlist is implemented using a double linked list.

The app provides an ADD SONG option.

When users select the ADD SONG option, a new node is added to the end of the playlist.

An example playlist is shown below. The head of this double linked list points to the node stored at memory location 218.



Describe the operations carried out on the double linked list when selecting the ADD SONG option.

3

2. A program is used to process the details of 5000 marathon runners. Sample runner details are shown below.

runnerName	runnerNumber	timeMins
Ryan Smith	2461	242
Graham Philips	1892	233
Ruth Sturrock	3781	207
Elizabeth Ferry	196	275

This program processes the details using an array of records based on the following record structure and array declaration.

```
RECORD Runner IS {STRING runnerName, INTEGER runnerNumber,
INTEGER timeMins}
```

```
DECLARE athletes AS ARRAY OF Runner INITIALLY [{runnerName =
"name", runnerNumber = 0, timeMins = 0}] * 5000
```

The incomplete pseudocode below uses the bubble sort algorithm to sort the `athletes` array into ascending order of time.

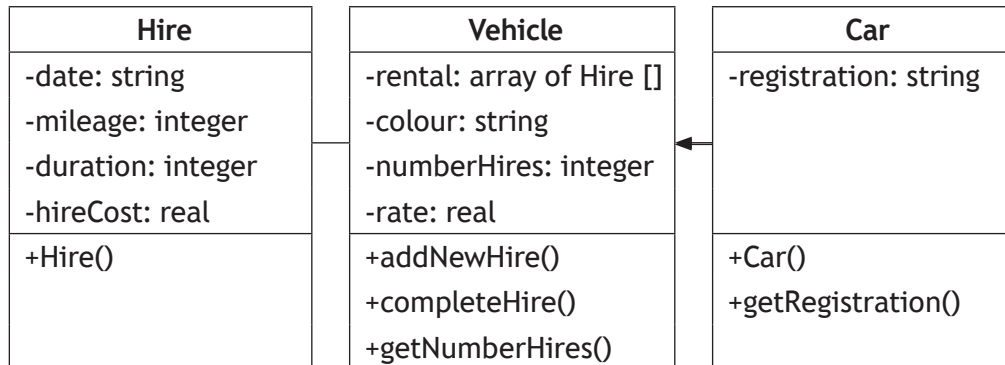
1. set `n = 4999`
2. start conditional loop
3. set `swapped = false`
4. start fixed loop `i = 0` to `n-1`
5. if _____ > _____ then
6. swap runner details
7. set `swapped = true`
8. end if
9. end fixed loop
10. set `n = n - 1`
11. end conditional loop when `swapped = false`

- (a) Write the condition needed at line 5 of the algorithm. 2
- (b) Explain why the variable `n` has been used in this sort. 1
- (c) An evaluation of the program would consider the efficiency of the code.
 Comment on the efficiency of the bubble sort algorithm used to sort the `athletes` array into ascending order of time. 2

[Turn over

3. GlasgowGreen uses an object-oriented program to track the use of hire vehicles across the city.

A simplified UML class diagram for the program is shown below.



Some of the object-oriented code used in the program is shown below.

```

Line 1  CLASS Vehicle IS { ARRAY OF Hire rental,  STRING colour,
        INTEGER numberHires, REAL rate }
Line 2      METHODS
Line 3      PROCEDURE addNewHire(STRING date)
Line 4          SET THIS.rental[THIS.numberHires].date TO date
Line 5          SET THIS.rental[THIS.numberHires].mileage TO 0
Line 6          SET THIS.rental[THIS.numberHires].duration TO 0
Line 7          SET THIS.rental[THIS.numberHires].hireCost TO 0.0
Line 8      END PROCEDURE

Line 9      PROCEDURE completeHire (INTEGER mileage, INTEGER
        duration)
Line 10          SET THIS.rental[THIS.numberHires].duration TO
        duration
Line 11          SET THIS.rental[THIS.numberHires].mileage TO
        mileage
Line 12          SET THIS.rental[THIS.numberHires].hireCost TO
        THIS.rate * THIS.rental[THIS.numberHires].mileage
Line 13          SET THIS.numberHires TO THIS.numberHires + 1
Line 14      END PROCEDURE

...
Line 23  END CLASS
  
```

3. (continued)

```

Line 24  CLASS Car INHERITS Vehicle WITH { STRING registration }
Line 25      METHODS
Line 26      CONSTRUCTOR ( STRING registration, STRING colour )
Line 27          DECLARE THIS.registration INITIALLY registration
Line 28          DECLARE THIS.rate INITIALLY 7.25
Line 29          DECLARE THIS.rental AS ARRAY OF Hire INITIALLY[]
Line 30          DECLARE THIS.colour INITIALLY colour
Line 31          DECLARE THIS.numberHires INITIALLY 0
Line 32      END CONSTRUCTOR
...

```

A `Car` object is instantiated using the following statement.

```
DECLARE car1 AS Car INITIALLY Car ("ABC123", "Red")
```

- (a) State what is stored in all of the instance variables of the `car1` object upon instantiation. 1
- (b) Using appropriate object-oriented terminology, explain why the following statement would be invalid. 2

```
SET car1.registration TO "DEF456"
```

- (c) Vehicle hire across the city has proved to be very popular and GlasgowGreen has decided to expand the business by introducing pedal bikes for hire.

A second subclass of the `Vehicle` class called `Bike` must be added to the program.

- (i) Explain why the use of subclasses with inheritance will reduce the time needed to add the `Bike` class to the program. 1
- (ii) The `Bike` class will have one additional private property, `bikeID`, which is a string value, and three additional public methods called `Bike()`, `completeHire()` and `getBikeID()`.

Complete the UML class diagram to include the `Bike` class, showing all properties and methods. You need only display the name of each class in the existing UML class diagram. 2

[Turn over

3. (c) (continued)

- (iii) The variable `vehicleArray` is an array of `Vehicle` objects. This array stores the details of all the vehicles owned by `GlasgowGreen`.

An object of the `Car` class is stored in the `vehicleArray` in position 20.

Using appropriate object-oriented terminology, explain why the following codes returns an error.

```
SET carReg TO vehicleArray[20].getRegistration()
```

3

- (d) The variable `bikeArray` is an array of `Bike` objects. This array stores the details of the 125 bikes owned by `GlasgowGreen`.

`GlasgowGreen` wants to know the ID of the bike that has been hired most often.

Using a programming language of your choice, write the code to find and display the ID of the bike that has been hired most often.

3

4. A block of flats has several software-controlled lifts to access all eight floors of the building. Users of the lifts have complained about waiting times. The developers of the existing control software have been asked to update the software to reduce waiting times for lift users.

- (a) (i) Time estimates for the analysis and design stages of this project are listed below:

- carry out feasibility study — 2 days
- produce requirements specification — 3 days
- design of data structures — 2 days
- design of software logic — 4 days.

Both stages of design should take place at the same time.

Draw a Gantt chart for the analysis and design stages of the project.

2

- (ii) Explain why economic and technical feasibility must be considered before proceeding with any project.

2

- (b) Lifts in the building may be descending, ascending or stationary. In the example below, Lift 1 is ascending, Lift 3 and Lift 4 are descending and Lift 2 is stationary.

	Lift 1	Lift 2	Lift 3	Lift 4
7th Floor				
6th Floor				
5th Floor	A		D	
4th Floor				
3rd Floor				
2nd Floor				D
1st Floor				
Ground		S		

A = Ascending

D = Descending

S = Stationary

One possible implementation of the updated control software would make use of a 2-D array.

- (i) Using a programming language of your choice, declare a 2-D array called `lifts` to store the lift details.
- (ii) Write code to assign the lifts to the four locations indicated above.

2

1

[Turn over

4. (b) (continued)

- (iii) Code will be used to find the nearest lift for any floor whenever the up or down button is pressed by the user.

The user inputs will be stored in two variables, `floor` and `direction`. Details of these variables are provided in the table below.

Variable Name	Possible Values	Purpose
<code>floor</code>	0 to 7	indicates floor user is on when the button is pressed
<code>direction</code>	UP or DOWN	indicates whether the user wants to go up or down

The software will calculate how far away all suitable lifts are from the user's floor and store these results in a 1-D array called `distance`. Unsuitable lifts will be assigned the value 99.

Using the lift details in part (b), when a user on the third floor presses the DOWN button the following logic is applied.

Lift	Suitability	Reason	Assigned value
1	Unsuitable	above the user and travelling upwards	99
2	Suitable	three floors below the user and stationary	-3
3	Suitable	two floors above the user and descending	2
4	Unsuitable	below the user and travelling downwards	99

These results would be stored in a 1-D array called `distance`, as shown below.

<code>distance</code>	99	-3	2	99
-----------------------	----	----	---	----

Using pseudocode, design an algorithm to calculate and store the correct results in the `distance` array for a single user, on any floor, travelling in either direction.

4

- (iv) Using pseudocode, design an algorithm that will find and display the number of the nearest lift using the values in the `distance` array.

2

- (c) Describe the difference between usability testing with prototypes and end-user testing.

2

[END OF SECTION 1]

SECTION 2 — DATABASE DESIGN AND DEVELOPMENT — 20 marks

Attempt ALL questions

5. The structure of a relational database is shown below.

Item	Order	OrderItem	Customer
<u>itemID</u>	<u>orderID</u>	<u>orderID *</u>	<u>customerNo</u>
itemName	orderDate	<u>itemID *</u>	customerName
price	customerNo *	<u>date</u>	email
stock		quantity	password

A sample record from the `Order` table is shown below.

orderID	orderDate	customerNo
245427	2021-05-04	SHG46

The following restrictions apply to the `Order` table:

- A unique `orderID` is automatically generated by the system whenever a new order is received.
- When customers open an account with the company, they are allocated a unique `customerNo` that consists of 5 characters.
- All orders must have an `orderDate`.

Write the SQL statement used to create the structure of the `Order` table.

3

6. EcoVenture is a walking club. The club website allows registered walkers to record details about walking routes they complete.

Data about the walkers and their completed walking routes is stored in a database with four related tables. The attributes in each table are listed below.

Route	Walker	Hill	Walk
<u>routeID</u>	<u>walkerID</u>	<u>hillID</u>	<u>walkerID*</u>
routeName	walkerName	hillName	<u>routeID*</u>
routeLength	walkerExperience	hillType	<u>walkDate</u>
routeDifficulty	walkerDOB	hillHeight	walkStartTime
hillID*			walkDistance
			walkEndTime
			walkDifficulty

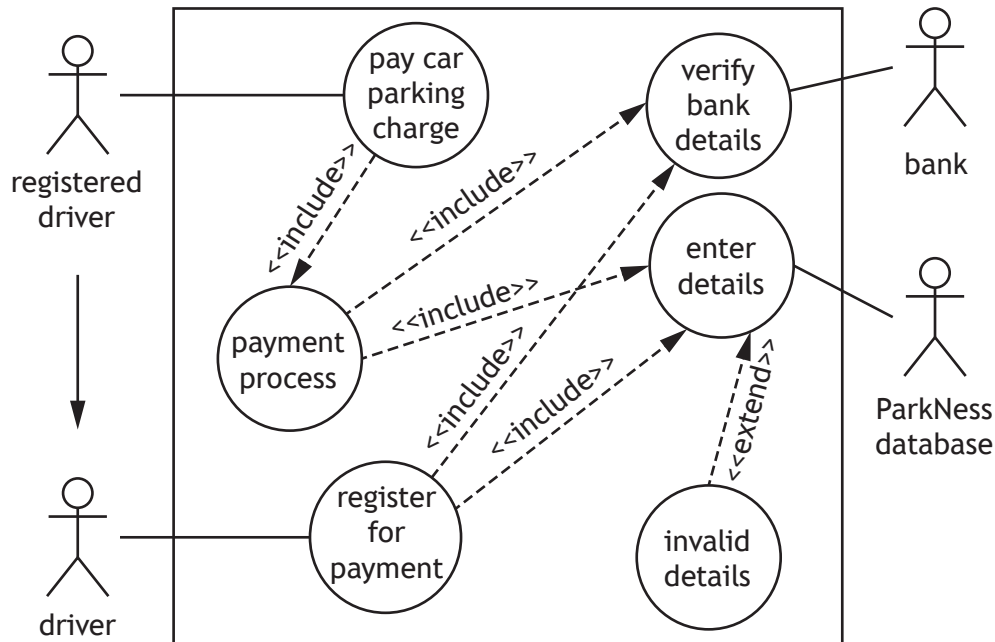
Explain why the `Walk` table would benefit from the introduction of a surrogate key.

2

7. ParkNess manages six car parks in Inverness city centre using a database-driven website.

Drivers can use the ParkNess app to pay for their parking by providing the licence plate of their car and details of their bank payment card.

The UML use case diagram below illustrates the functionality of the ParkNess app.



- (a) (i) Describe the relationship between the actors 'driver' and 'registered driver'.
- (ii) Describe the use of the 'extend' relationship between the 'enter details' and 'invalid details' use cases.

1

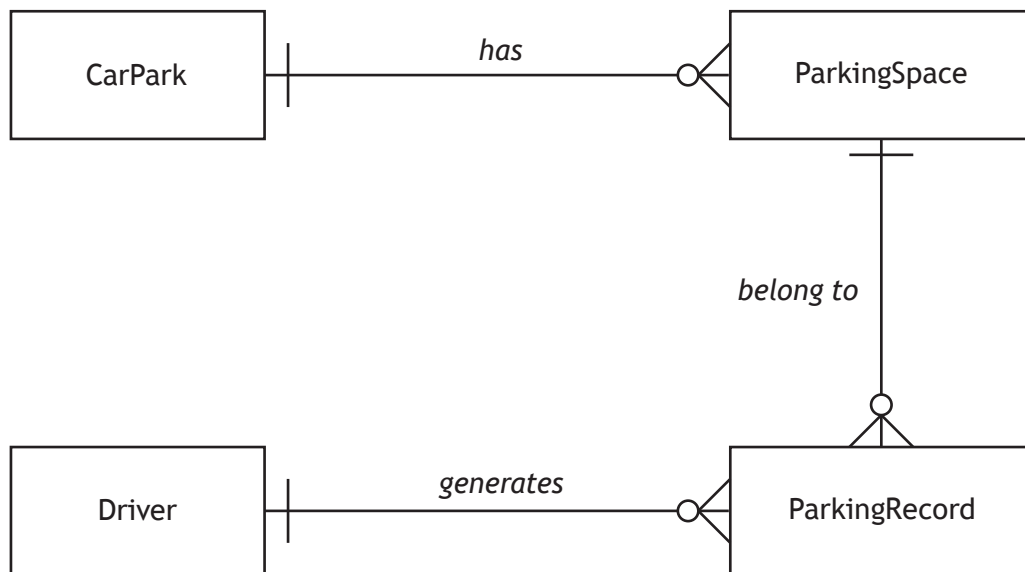
1

7. (continued)

ParkNess makes use of Automatic Number Plate Recognition to record the licence plates of parked cars. Each parking space has a sensor that records whether it is occupied. The records for every parking space in the car parks are stored in a relational database. The structure of the database is shown below.

CarPark	ParkingSpace	ParkingRecord	Driver
<u>carParkID</u>	<u>spaceID</u>	licencePlate	<u>driverID</u>
carParkName	carParkID *	<u>entryDate</u>	email
capacity	occupied	<u>entryTime</u>	paymentCardNumber
		departureTime	expiry
		<u>spaceID *</u>	
		cost	
		<u>driverID *</u>	

(b) The entity-relationship diagram for this database is shown below.



- State whether each entity in this entity-relationship diagram is strong or weak.
- Describe the relationship participation between the `ParkingSpace` and `ParkingRecord` entities.

2

2

[Turn over

7. (continued)

Sample data to be stored in each table of the database is shown below.

CarPark		
carParkID	carParkName	capacity
cp1	Rose Street	95
cp2	Telford Street	100
cp3	Bank Street	150
cp4	Clachnaharry Road	80
cp5	Ness Walk	64
cp6	Stadium Road	144

ParkingSpace		
spaceID	carParkID	occupied
33687	cp4	0
41722	cp2	1
...

*Note that a value of 0 in the **occupied** field indicates that the space is unoccupied.*

Driver			
driverID	email	paymentCardNumber	expiry
145	stevenm@inverness.co.uk	1111222233334444	08/21
158	daveb@ness.com	2222111133334444	09/21
193	melanieph@north.com	3333444411112222	10/21
...

ParkingRecord						
licencePlate	entryDate	entryTime	departureTime	spaceID	cost	driverID
SM68 JIX	2021-04-08	08:55	15:18	33687	10.50	145
SS16 PVM	2021-04-09	11:12	19:50	54256	13.50	158
DB19 SJS	2021-04-07	18:08	20:19	12346	4.50	193
SM68 JIX	2021-04-09	09:12	16:22	36027	12.00	145
SJ19 DBM	2021-04-07	09:43	17:02	12346	12.00	127
MI17 BSS	2021-04-09	07:22	18:17	65431	13.50	149
GC15 SNE	2021-04-08	11:10	13:18	12346	4.50	137
...

(c) Describe the output produced from the following query.

1

```
SELECT spaceID
FROM ParkingSpace
WHERE NOT carParkID = "cp4"
AND occupied = 0;
```

7. (continued)

- (d) (i) The SQL query below is used to display details of those drivers who parked in a ParkNess car park at least 10 times during April 2021.

```
SELECT email, licencePlate, COUNT(*) AS [Number of  
Occasions]  
FROM Driver, ParkingRecord  
WHERE Driver.driverID = ParkingRecord.driverID  
AND _____  
GROUP BY email, licencePlate  
_____ ;
```

Using appropriate Advanced Higher operators, write the missing conditions needed to complete this query.

2

- (ii) Andrew works near the Rose Street and Bank Street car parks.

The SQL query below is used to show the name of these two car parks along with the number of available spaces in each car park.

```
SELECT carParkName, capacity - COUNT(*) AS [Spaces  
Available]  
FROM CarPark, ParkingSpace  
WHERE CarPark.carParkID = ParkingSpace.carParkID  
AND _____  
GROUP BY carParkName;
```

Using the `IN` operator, write a condition needed to complete the `AND` clause of this query.

1

[Turn over

- (e) ParkNess uses a SQL query to display details of individual parking spaces from all car parks other than the Ness Walk car park.

Each parking space displayed by the query has a total income which is more than twice the income generated from all the parking spaces in the Ness Walk car park that were used between 14 and 26 April 2021 inclusive.

Complete the design of this query by writing the missing parts labelled A, B and C of the query design below.

3

Field(s) / Calculation(s)		spaceID, carParkName, totalIncome = sum(cost)		
Table(s)		CarPark, ParkingSpace, ParkingRecord		
Search Criteria		A		
Grouping		spaceID, carParkName		
Having	B	Inner Query	Field(s) / Calculation(s)	C
			Table(s)	ParkingSpace, ParkingRecord, CarPark
			Search criteria	entryDate between 2021-04-14 and 2021-04-26
			AND	carParkName = "Ness Walk"
Sort Order		spaceID ASC		

- (f) Before using the car park for the first time, drivers have to register using an online form.

An extract of the HTML form code is shown below.

```
<input type="text" id="driverID" name="driverID" required>
<input type="email" id="email" name="email" required>
<input type="text" id="paymentCardNumber"
name="paymentCardNumber" required>
<input type="text" id="expiry" name="expiry" required>
<input type="submit" value="Submit"> </form>
```

- (i) Explain the use of `required` in the code above.
- (ii) Describe how code would be used to ensure the `paymentCardNumber` is exactly 16 characters long.

1

1

[END OF SECTION 2]

SECTION 3 — WEB DESIGN AND DEVELOPMENT — 20 marks

Attempt ALL questions

8. A dynamic website can display statistical information based on historical weather data collected from a weather station.

Weather					
year	month	maxTemp	minTemp	rainfall	sunHours
2018	1	6	1.2	175.1	39.7
2018	2	6.4	0.1	67.6	94.5
2018	3	7.2	1.3	70.4	87.4
2018	4	12	5.2	70.1	116
2018	5	17.7	7.8	27.6	232.8
2018	6	20.9	10.5	59	252.4
2018	7	21.6	12.7	60	218.8
2018	8	18.1	11.5	81.7	110.6
2018	9	15.5	8.9	100.6	114.9
2018	10	13	6.2	104.9	94.1
2018	11	10.1	4.7	148.1	46.6
2018	12	12	-3.1	98.3	28

Draw the output generated by the following server-side code.

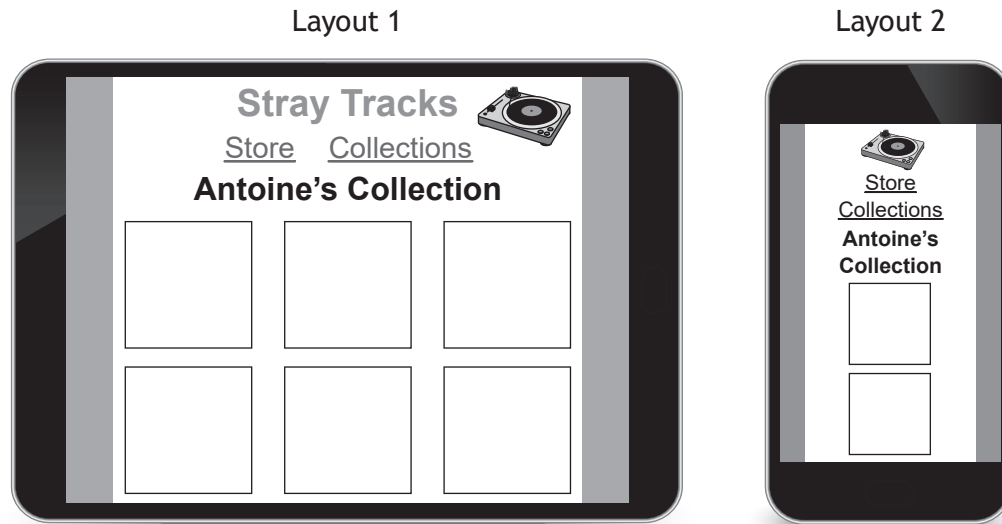
2

```
$sql = "SELECT month, sunHours FROM Weather WHERE sunHours
BETWEEN 200 AND 250";
$result = mysqli_query($conn, $sql);
echo "<table><tr><th> Month </th><th> Hours of Sunshine
</th></tr>";
while($row = mysqli_fetch_array($result)) {
    echo "<tr>
        <td>" . $row['month'] . "</td>
        <td>" . $row['sunHours'] . "</td>
    </tr>";
}
echo "</table>";
```

[Turn over

9. A new website called Stray Tracks allows music fans to store and share details of their music collections with other users of the website.

When browsing a user's music collection, a gallery of album covers is displayed. The gallery layout is controlled using a media query.



Some of the incomplete CSS and HTML code used to produce these layouts is shown below.

CSS

```
...
body{margin:auto;}
nav {height:35px;}
h1 {display:inline; font-size:28px;}
nav ul li {width:150px; height:20px; font-size:16pt;
float:left;}
@media screen and (max-width: 460px) {
    nav {height:70px;}
    nav ul li {width:100%; font-size:8pt;}
    ...
}
```

HTML

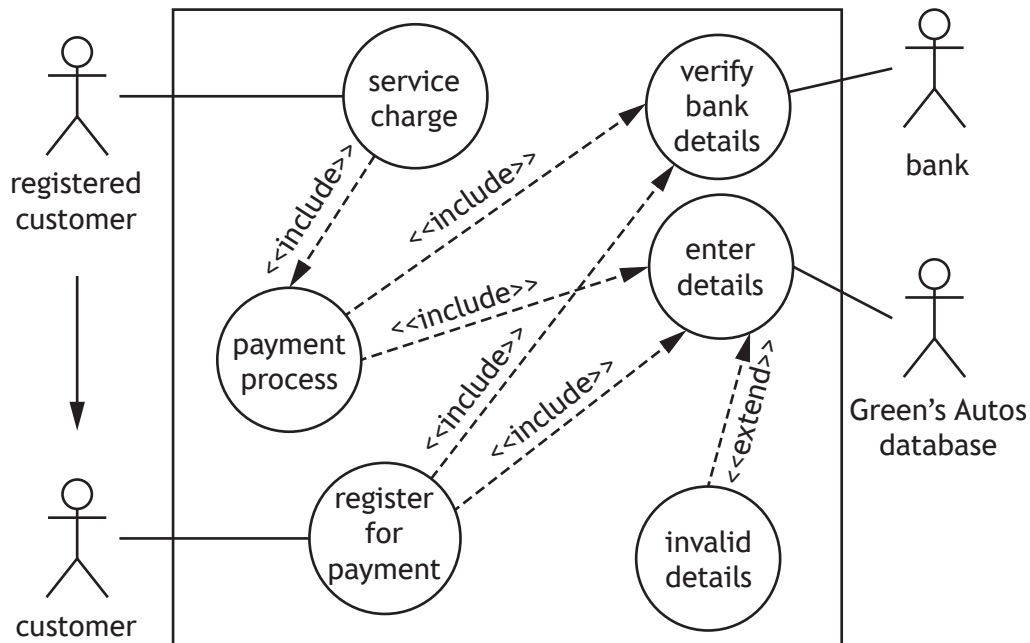
```
<header><h1 id="mainHeading"> Stray Tracks </h1></header>
<nav><ul><li> <a href="store.html"> Store </a></li>
<li><a href="collections.html"> Collections </a></li></ul>
</nav>
```

- (a) Describe when the media query would be used to generate Layout 2. 1
- (b) Explain why the `nav height` value is larger in the media query than in the rule `nav {height:35px;}`. 1
- (c) Explain, making reference to the main heading, Stray Tracks, why the CSS is incomplete. 1

10. Green's Autos offers a wide range of cars for sale and provides a range of car servicing, testing and repairs. They manage this using a database-driven website.

- (a) Customers can pay for any servicing carried out using the company's database-driven website by providing the licence plate of their car and details of their bank payment card.

The UML use case diagram below illustrates the functionality of the car service payment facility.



- (i) Describe the relationship between the actors 'customer' and 'registered customer'.
- (ii) Describe the use of the 'extend' relationship between the 'enter details' and 'invalid details' use cases.

1

1

[Turn over

10. (continued)

- (b) Green's Autos website also allows customers to search for details of cars for sale. The page below shows the results of one such search.

Car Sales
<div><h2>Green's Autos</h2><h3>Yotoya Rocrolla</h3><p>£12,999</p><p>Request a test drive</p></div>

To request a test drive for the car being viewed, the customer clicks the 'Request a test drive' button. This loads a new page with the online form shown below. The customer then provides each of the details in the form.

Car Sales
<div><h2>Green's Autos</h2><h3>Request a test drive</h3><p>Name <input type="text"/></p><p>Email <input type="text"/></p><p>Date <input type="text" value="yyyy-mm-dd"/></p><p>Make Yotoya</p><p>Model Rocrolla</p><p>Submit details</p></div>

- (i) Write the HTML `<form>` statement used to submit the form details securely to the server-side script called `requestConfirm.php`.

10. (b) (continued)

- (ii) Write the HTML statement used to generate the part of the form shown below.

1

Date	<input type="text" value="yyyy-mm-dd"/>
------	---

- (c) (i) The PHP script `requestConfirm.php` generates a separate 'Confirm Page' of the website shown below. This is used to confirm that the customer's test drive request has been received.

Car Sales
<p style="text-align: center;">Green's Autos</p> <p style="text-align: center;">Thank you</p> <p>Dear Melanie Peters</p> <p>Thank you for requesting a test drive of a Yotoya Rocrolla.</p> <p>We will check availability on 22/06/2021 and send an email to mp27@email.com to confirm your booking.</p> <p>Click below to return to the car page.</p> <p style="text-align: center;">Return to car page</p>

Explain the term session variables, making reference to the page above.

2

- (ii) The PHP script `requestConfirm.php` also forms a connection with the secure database server by performing the following tasks.

- (1) Assign the connection details to server-side variables
- (2) Create a connection called `$conn` with the secure database

Write the PHP code required to perform these two tasks. Your code should make use of the connection details below:

- server name = `serv1a`
- username = `root`
- password = `AQS20`
- database = `TestDrive`

2

10. (continued)

- (d) (i) Once a connection has been created, details of a customer's test drive request are assigned to three variables:

- `$customerName`
- `$email`
- `$startDate`

The values in these variables will then be stored in a database table called `ClientTD48`. The structure of this table is shown below.

Attribute	Type	Size	Key	Required	Validation
<code>id</code>	<code>int</code>	4	PK	Y	Auto increment
<code>customerName</code>	<code>varchar</code>	50		Y	
<code>email</code>	<code>varchar</code>	30		Y	
<code>startDate</code>	<code>date</code>			Y	
<code>requestStatus</code>	<code>varchar</code>	12		Y	Restricted choice: one of pending, successful or unsuccessful

A single query is used to:

- store the details of a customer's test drive request in a new record of the `ClientTD48` table
- assign the initial value 'pending' to the `requestStatus` field of this new record.

Write the SQL statement for this query.

2

10. (d) (continued)

(ii) The variable `$result` is generated. It contains the following data.

id	customerName	email	startDate	requestStatus
1734	Sylvan Broad	brs@email.com	2021-06-22	pending
1735	Mo Ali	ma13@me.com	2021-06-22	pending
1736	Joanne Leach	lj12@email.com	2021-06-22	pending
1737	Melanie Peters	mp27@email.com	2021-06-22	pending
1738	Thomas Murtagh	murt@tmail.com	2021-06-22	pending
1739	Marge Smith	ma14@imail.com	2021-06-22	pending
1740	Scott Curran	scus@me.com	2021-06-22	pending

A maximum of five test drives is permitted on any given date.

Server-side code is used to set the `requestStatus` field of the `ClientTD48` table to 'successful' for the first five requests for any given date or 'unsuccessful' for all additional requests.

Using pseudocode, design an algorithm to process the `$result` variable and amend the `requestStatus` field of each relevant record in the `ClientTD48` table appropriately.

3

(e) All stored session data associated with a test drive request should be deleted as soon as the user has completed the session.

State how this can be achieved.

1

[END OF SECTION 3]

[END OF SPECIMEN QUESTION PAPER]

Acknowledgement of copyright

Question 9 Image is taken from PNGkey. Reproduced under Creative Commons.

Question 10 Editorial Credit: Jose Gil/shutterstock.com



National
Qualifications
SPECIMEN ONLY

S816/77/11

Computing Science

Marking Instructions

These marking instructions have been provided to show how SQA would mark this specimen question paper.

The information in this publication may be reproduced to support SQA qualifications only on a non-commercial basis. If it is reproduced, SQA should be clearly acknowledged as the source. If it is to be used for any other purpose, written permission must be obtained from permissions@sqa.org.uk.

Where the publication includes materials from sources other than SQA (ie secondary copyright), this material should only be reproduced for the purposes of examination or assessment. If it needs to be reproduced for any other purpose it is the user's responsibility to obtain the necessary copyright clearance.

General marking principles for Advanced Higher Computing Science

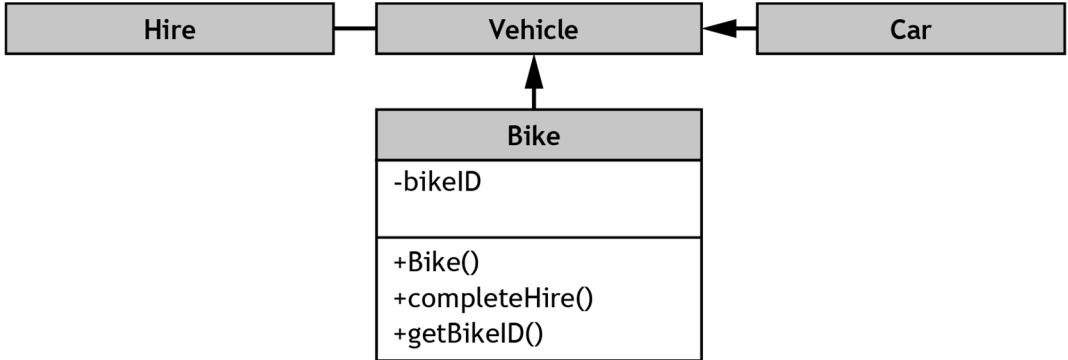
This information is provided to help you understand the general principles you must apply when marking candidate responses to questions in this paper. These principles must be read in conjunction with the detailed marking instructions, which identify the key features required in candidate responses.

- (a) Marks for each candidate response must **always** be assigned in line with these general marking principles and the detailed marking instructions for this assessment.
- (b) Always use positive marking. This means candidates accumulate marks for the demonstration of relevant skills, knowledge and understanding; marks are not deducted.
- (c) If a candidate response is not covered by either the principles or detailed marking instructions, and you are uncertain how to assess it, you must seek guidance from your team leader.
- (d) Award marks regardless of spelling, as long as the meaning is unambiguous. This applies to all responses, including code. Award marks as per the detailed marking instructions, regardless of syntax errors, if the intention of the coding is clear.
- (e) For questions where candidates are asked to design or write code, a sample response is 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.
- (f) If a candidate scores through a response 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.
- (g) Where an incorrect response is carried forward and used correctly in a following part of the question, you should give credit for subsequent responses that are correct with regard to the original error. Candidates should not be penalised more than once for the same error.
- (h) Only award marks for a valid response to the question asked. Where candidates are asked to:
 - **Identify, name, give or state**, they need only name or present in brief form.
 - **describe**, they must provide a statement or structure of characteristics and/or features. This will be more than an outline or a list. It may refer to, for example, a concept, process, experiment, situation, or facts, in the context of and appropriate to the question. Candidates must make the same number of factual/appropriate points as there are marks available in the question.
 - **explain**, they must relate cause and/or effect and/or make relationships between things clear, in the context of the question or a specific area within the question.
 - **write code**, they must write recognisable code, not prose nor a diagram.
 - **design**, they must use a design technique appropriate to the problem. Award marks as per the detailed marking instructions, regardless of errors in the exemplification of the technique, if the intention of the design is clear.
- (i) In the detailed 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.

Marking instructions for each question

Section 1 - Software design and development

Question			Expected response	Max mark	Additional guidance
1.			ADD SONG adds a new node at the end of the linked list. The next pointer of the last item in the playlist, the node at memory location 367, will be changed from NULL to point at the new node. The next pointer of the new node will be set to NULL and its previous pointer will be set to 367.	3	<p>1 mark for correct updating of the next pointer of the node at memory location 367</p> <p>1 mark for correct use of NULL as the next pointer of the new node</p> <p>1 mark for correct use of memory location 367 as the previous pointer of the new node</p>
2.	(a)		IF runners[i].finishTime > runners[i+1].finishTime THEN	2	<p>1 mark for comparing i with i+1</p> <p>1 mark for correct use of data structure</p> <p>Accept alternative comparison: IF runners[i+1] < runners[i] THEN</p>
	(b)		<p>n is used to control the size of the inner loop which depends on the number of items that are still to be processed</p> <p>OR</p> <p>n is decreased after a value has been placed in the correct position at the end of the array</p>	1	<p>1 mark each for either:</p> <ul style="list-style-type: none"> controls of inner loop which is used to process unsorted items counts down from sorted end of the array
	(c)		<p>The algorithm makes use of the Boolean swapped which will terminate the outer loop as soon as a fully sorted list is detected. Use of n to control the inner loop reduces the number of comparisons needed during each pass. On average, the bubble sort will make n^2 comparisons - in this case, since $n = 5000$, 25,000,000 comparisons would be needed.</p>	2	<p>1 mark each for a maximum of 2 marks:</p> <ul style="list-style-type: none"> use of Boolean swapped use of n to control inner loop reference to average number of comparisons needed

Question			Expected response	Max mark	Additional guidance
3.	(a)		registration = "ABC123" rate = 7.25 rental = empty array of Hire colour = "red" numberHires = 0	1	1 mark for all 5 correct Accept rental array is empty
	(b)		Encapsulation means that a property cannot be edited directly. A method must exist in order to update the registration property.	2	1 mark for use of encapsulation or use of a private property 1 mark for need for method
	(c)	(i)	Methods from the superclass can be used which avoids the need to code them from scratch	1	1 mark for reuse of methods already defined in the superclass
		(ii)	See below.	2	1 mark for <code>Bike</code> class with correct properties and methods 1 mark for inheritance indicated correctly
			 <pre> classDiagram Hire --> Vehicle Car -- > Vehicle Bike -- > Vehicle class Bike { -bikeID +Bike() +completeHire() +getBikeID() } </pre>		
		(iii)	<p>This is an example of polymorphism since the variable <code>vehicleArray</code> belongs to the superclass <code>Vehicle</code> and can be associated with any objects that belong to its subclasses.</p> <p>Since the variable <code>vehicleArray</code> belongs to the <code>Vehicle</code> class, it can only access methods that are associated with the superclass and all elements contained within the array must follow that principle.</p> <p>Although the <code>getRegistration()</code> method is associated with the <code>Car</code> class, it is not associated with the superclass and this attempt to access the method will fail.</p>	3	1 mark for explanation of the use made of polymorphism 1 mark for stating that the <code>vehicleArray</code> belongs to the superclass 1 mark for stating that the <code>getRegistration()</code> method is only associated with the <code>Car</code> class and is not accessible to superclass variables

Question			Expected response	Max mark	Additional guidance
3.	(d)		See below.	3	<p>1 mark for correct use of bikeArray</p> <p>1 mark for use of getNumberHires() method</p> <p>1 mark for use of getRegistration() method</p> <p>Alternative solutions are possible</p>
			<pre> DECLARE maxPosition INITIALLY 0 FOR loop FROM 0 TO 124 DO IF bikeArray[loop].getNumberHires() > bikeArray[maxPosition].getNumberHires() THEN SET maxPosition TO loop END IF END FOR SEND bikeArray[maxPosition].getRegistration() TO DISPLAY </pre>		

Question			Expected response	Max mark	Additional guidance																																																		
4.	(a)	(i)	See Gantt chart below.	2	1 mark for Feasibility Study, Requirements Specification and Data Structure Design happening sequentially. 1 mark for overlapping Data Structure Design and Software Logic Design.																																																		
			<table><tr><td></td><td>1</td><td>2</td><td>3</td><td>4</td><td>5</td><td>6</td><td>7</td><td>8</td><td>9</td></tr><tr><td>Feasibility Study</td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td></tr><tr><td>Requirements Specification</td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td></tr><tr><td>Data Structure Design</td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td></tr><tr><td>Software Logic Design</td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td></tr></table>				1	2	3	4	5	6	7	8	9	Feasibility Study										Requirements Specification										Data Structure Design										Software Logic Design									
	1	2	3	4	5	6	7	8	9																																														
Feasibility Study																																																							
Requirements Specification																																																							
Data Structure Design																																																							
Software Logic Design																																																							
		(ii)	If the cost of creating the software upgrade is prohibitively high and if the technical abilities of the developers/hardware requirements for the solution are not suitable, it will not be feasible to continue with the development work.	2	1 mark for economic feasibility 1 mark for technical feasibility																																																		
	(b)	(i)	SET lifts TO [[]] * 8 FOR count FROM 0 TO 7 DO SET lifts[count] TO [""] * 4 END FOR OR DECLARE lifts AS ARRAY OF ARRAY OF CHARACTER INITIALLY < 4 x 8 array, all set to blank >	2	1 mark for dimensions 1 mark for data type Dimensions of 2-D array must be indicated. Data type must be indicated explicitly as CHARACTER or implied using "" for null string.																																																		
		(ii)	SET lifts [2] [0] TO "A" SET lifts [7] [1] TO "S" SET lifts [2] [2] TO "D" SET lifts [5] [3] TO "D"	1	1 mark for correct assignment of all values Accept answers where the bottom row is numbered from 0																																																		

Question			Expected response	Max mark	Additional guidance
4.	(b)	(iii)	See below.	4	<p>1 mark for nested loop to process 2-D array.</p> <p>1 mark for finding element with lift in each column.</p> <p>1 mark for deciding if lift will not be called (assign 99 to distance).</p> <p>1 mark for calculating and storing difference if lift can be called.</p>
			<pre> 1. FOR row FROM 0 TO 7 DO 2. FOR column FROM 0 TO 3 DO 3. IF lifts [row] [column] <> "" THEN 4. IF lifts [row] [column] = "S" THEN 5. SET distance[column] TO floor - row 6. ELSE IF direction = "A" AND row > floor THEN 7. SET distance[column] TO floor - row 8. ELSE IF direction = "D" AND row < floor THEN 9. SET distance[column] TO floor - row 10. ELSE 11. SET distance[column] TO 99 12. END IF 13. END IF 14. END FOR 15. END FOR </pre>		
		(iv)	See below.	2	<p>1 mark for dealing with negative values</p> <p>1 mark for finding correct lift</p>
			<pre> 1. loop for each value in distance array 2. if value <0 then 3. convert to a positive value 4. end if 5. end loop 6. set position of minimum = 0 7. loop for each lift 8. if distance < minimum then 9. update position of minimum 10. end if 11. end loop 12. display lift number </pre> <p><i>Note: a pre-defined function may be used to find magnitude and ignore sign of each value. Or negative values can be made positive inside finding minimum algorithm.</i></p>		

Question			Expected response	Max mark	Additional guidance
4.	(c)		<p>Usability testing with prototypes would be conducted at an early stage in development, possibly with the end-user, using manual demonstrations to show how software will work, (with the development team making notes on end-user feedback.)</p> <p>End-user testing would be conducted at the end of the development, with the end users of the software using a live version of the solution, (with the development team making notes on end-user feedback.)</p>	2	<p>1 mark for usability testing with prototypes</p> <p>1 mark for end-user testing</p>

Section 2 - Database design and development

Question			Expected response	Max mark	Additional guidance
5.			See below.	3	1 mark for CREATE TABLE with 3 fields and correct data types 1 mark for primary key correct 1 mark for foreign key correct
			<pre>CREATE TABLE Order(orderID int AUTO-INCREMENT PRIMARY KEY, orderDate date NOT NULL, customerID varchar(5) NOT NULL, FOREIGN KEY (customerID) REFERENCES Customer(customerID));</pre>		
6.			<p>A surrogate key could be used to replace the current primary key which is a compound key.</p> <p>Use of a surrogate key would make it easier to reference historical data.</p> <p>It would also improve the performance of queries that involve a join with the Walk table.</p> <p>In this situation, it would allow a walker to record completion of a route more than once on the same day.</p>	2	<p>1 mark for purpose of the surrogate key</p> <p>1 mark for one appropriate benefit</p>

Question			Expected response	Max mark	Additional guidance
7.	(a)	(i)	The 'registered driver' actor inherits all the characteristics of the 'driver' actor.	1	1 mark for inheritance relationship
		(ii)	Extends indicates a conditional use case. In this situation, the 'enter details' use case may lead to the 'invalid details' use case - but it may not.	1	1 mark for description that refers to the conditional nature of the relationship
	(b)	(i)	CarPark and Driver are strong entities ParkingSpace and ParkingRecord are weak entities	2	1 mark for both strong entities 1 mark for both weak entities
		(ii)	Each parked car in the ParkingRecord entity must have a corresponding parking space in the ParkingSpace entity so the ParkingRecord participation is mandatory. A ParkingSpace may not be used so the ParkingSpace participation is optional.	2	1 mark for mandatory participation of ParkingRecord entity 1 mark for optional participation of ParkingSpace entity
	(c)		This would produce a list of parking spaces available in car parks 1, 2 and 3.	1	
	(d)	(i)	entryDate BETWEEN "2021-04-01" AND "2021-04-30" HAVING COUNT(*) >= 10	2	1 mark for correct use of BETWEEN (ignore incorrect date format) 1 mark for correct use of HAVING
		(ii)	AND carParkName IN ('Rose Street', 'Bank Street')	1	
	(e)		A: NOT carParkName = "Ness Walk" B: sum(cost) > C: sum(cost) x 2	3	1 mark for each correct
	(f)	(i)	required conducts a presence check on the field as input validation to ensure that data has been entered.	1	
		(ii)	Add: minlength=16 maxlength = 16 to the input element for paymentCardnumber	1	

Section 3 - Web design and development

Question			Expected response	Max mark	Additional guidance						
8.			<table><tr><td>Month</td><td>Hours of Sunshine</td></tr><tr><td>5</td><td>232.8</td></tr><tr><td>7</td><td>218.8</td></tr></table>	Month	Hours of Sunshine	5	232.8	7	218.8	2	1 mark for correct headings 1 mark for correct rows selected
Month	Hours of Sunshine										
5	232.8										
7	218.8										
9.	(a)		When the webpage is viewed on a display/device/browser window narrower than 460 pixels.	1							
	(b)		So that the hyperlinks can be displayed vertically rather than horizontally.	1							
	(c)		An additional rule must be added to the media query to tell the browser to hide the main heading.	1	1 mark for explanation of need for an additional rule (CSS rule not required). Accept correct CSS rule (without the explanation): #mainHeading{display:none}						
10.	(a)	(i)	The ‘registered customer’ actor inherits all the characteristics of the ‘customer’ actor.	1	1 mark for inheritance relationship						
		(ii)	Extends indicates a conditional use case. In this situation, the ‘enter details’ use case may lead to the ‘invalid details’ use case - but it may not.	1	1 mark for description that refers to the conditional nature of the relationship						
	(b)	(i)	<form method="post" action="requestConfirm.php" >	2	1 mark for correct action 1 mark for correct method						
		(ii)	Date <input type = "text" value = "yyyy-mm-dd">	1	1 mark for input element with correct use of the value attribute						
	(c)	(i)	Session variables allow values stored in variables to persist across several pages of the website. This means that the values can be used on multiple pages of the site. Session variables have been used on this page to store details of the car (make and model), customer (name and email).	2	1 mark for use made of session variables 1 mark for details of values being stored in session variables						

Question			Expected response	Max mark	Additional guidance
10.	(c)	(ii)	<pre>//assign connection details \$servername = "serv1a"; \$username = "root"; \$password = "AQS20"; \$dbase="TestDrive"; // create connection \$conn = mysqli_connect(\$servername, \$username, \$password,\$dbase);</pre>	2	<p>1 mark for assignment of connection details to server-side variables</p> <p>1 mark for assignment of \$conn that makes use of mysqli_connect function with 4 parameters</p>
	(d)	(i)	<pre>INSERT INTO ClientTD48 (customerName, startDate, email, requestStatus) VALUES ('\$customerName', '\$startDate', '\$email', "pending");</pre>	2	<p>1 mark for INSERT query with correct list of fields</p> <p>Note: id field should not be listed since it is an auto increment field</p> <p>1 mark for list of values to match field sequence</p>
		(ii)	<pre>1. set pending = 0 2. start conditional loop while there are still query results to process 3. if pending < 5 then 4. update requestStatus field of current record to 'successful' 5. else 6. update requestStatus field of current record to 'unsuccessful' 7. end if 8. add 1 to pending 9. end conditional loop</pre>	3	<p>1 mark for ensuring that only 5 records are successful</p> <p>1 mark for updating successful records</p> <p>1 mark for updating unsuccessful records</p> <p>Alternative solutions are possible</p>
	(e)		This could be achieved using the session_destroy() statement.	1	

[END OF SPECIMEN MARKING INSTRUCTIONS]

Published: October 2023

Change since last published:

Graphic for question 7 (b) amended.

Question 10 (d) (ii) reworded.