



National 5 Computing Science Assignment Assessment task

This document provides information for teachers and lecturers about the coursework component of this course in terms of the skills, knowledge and understanding that are assessed. It must be read in conjunction with the course specification.

Valid for session 2021-22 only.

This assessment is given to centres in strictest confidence. You must keep it in a secure place until it is used.

Contents

Introduction	1
Instructions for teachers and lecturers	2
Instructions for candidates	7

Introduction

This document contains instructions for teachers and lecturers, and instructions for candidates for the National 5 Computing Science assignment. You must read it in conjunction with the course specification.

This assignment has 40 marks out of a total of 120 marks available for the course assessment.

This is one of two course assessment components. The other component is a question paper.

Instructions for teachers and lecturers

This assessment applies to the assignment for National 5 Computing Science for the academic session 2021-22.

The task is valid for 2021-22 only. Once complete, you must send the assignment responses to SQA to be marked.

You must conduct the assignment under a high degree of supervision and control. This means:

- ◆ candidates must be supervised throughout the session(s)
- ◆ candidates must not have access to email or mobile phones
- ◆ candidates must complete their work independently – no group work is permitted
- ◆ candidates must not interact with each other
- ◆ with no interruption for targeted learning and teaching
- ◆ in a classroom environment

Time

Candidates have 6 hours to carry out the assignment, starting at an appropriate point in the course, after all content has been delivered. It is not anticipated that this is a continuous 6-hour session, although it can be, but conducted over several shorter sessions. This is at your discretion.

You have a responsibility to manage candidates' work, distributing it at the beginning and collecting it in at the end of each session, and storing it securely in between. This activity does not count towards the total time permitted for candidates to complete the assignment.

Candidates are prompted to print their work at appropriate stages of the tasks. They can print on an ongoing basis or save their work and print it later. Whatever approach they take, time for printing is not part of the 6 hours permitted for the assignment.

Resources

Each candidate must have access to a computer system with a high-level (textual) programming language and either:

- ◆ a database application or software that can create, edit and run SQL
- ◆ software that can create, edit and run HTML and CSS

This is an open-book assessment. Candidates can access resources such as programming manuals, class notes, textbooks and programs they have written throughout the course. These may be online resources.

You must not create learning and teaching tasks that make use of constructs required in the assessment task, **with the specific purpose of developing a solution that candidates can access during the assignment.**

There may be instances where restriction of network use is prohibited (for example, a local authority-managed network with specific limitations). However, it remains your professional responsibility to make every effort to meet the assessment conditions.

Reasonable assistance

The assignment consists of three independent tasks. They are designed in a way that does not require you to provide support to candidates, other than to ensure that they have access to the necessary resources. Candidates can complete the tasks in any order.

Once the assignment is complete, you must not return it to the candidate for further work to improve their mark. You must not provide feedback to candidates or offer an opinion on the perceived quality or completeness of the assignment response, at any stage.

You can provide reasonable assistance to support candidates with the following aspects of their assignments:

- ◆ printing, collating and labelling their evidence to ensure it is in the format specified by SQA
- ◆ ensuring candidates have all the materials and equipment required to complete the assignment – this includes any files provided by SQA
- ◆ ensuring candidates understand the conditions of assessment and any administrative arrangements around the submission and storage of evidence, and the provision of files
- ◆ technical support

Evidence

All candidate evidence (whether created manually or electronically) must be submitted to SQA in a paper-based format. The evidence checklist details all evidence to be gathered. You can use it to ensure you submit all evidence to SQA.

You should advise candidates that evidence, especially code, must be clear and legible. This is particularly important when pasting screenshots into a document.

There is no need for evidence to be printed single-sided or in colour.

Alteration or adaptation

The tasks are in PDF and Word formats. Each task is available as a separate file from the secure site. Word files allow candidates to word process their responses to parts of the task.

You must not adapt the assignment in any way that changes the instructions to the candidate and/or the nature and content of the tasks. However, you can make changes to font size, type and colour and to the size of diagrams for candidates with different assessment needs, for example, visual impairment.

If you are concerned that any particular adaptation changes the nature and/or the content of the task, please contact our Assessment Arrangements team for advice as soon possible at aarequests@sqa.org.uk.

Submission

Each page for submission has the number of the assignment task that it refers to, for example 1a, and contains space for candidates to complete their name and candidate number. Any other pages submitted, for example, prints of program listings or screenshots, must have this information added to them.

Specific instructions for teachers and lecturers: 2021-22

All candidates must complete task 1 (software design and development) and **either** task 2 (database design and development) or task 3 (web design and development).

It is at your discretion how you approach this optionality in assessment. The task your candidates complete might be pre-determined by your progress through the course, or you may be able to let candidates choose which task to complete.

You must follow these specific instructions and ensure that candidates are aware of what you will give them at each stage in the assessment.

Print each task on single-sided paper, where applicable, as this:

- ◆ allows candidates to refer to information on other pages
- ◆ helps you manage tasks that are split into more than one part

Task 1 – part A requires candidates to refine one process identified in the pseudocode. They must submit their evidence for this part of the task to you **before** starting part B, as this contains a possible solution to part A.

Task 1 – part B is a separate section. This ensures that candidates are not able to access part A and change their responses. Candidates must still have access to the program analysis page during part B.

Task 2 – part A requires candidates to analyse a database. They must submit their evidence for this part of the task **before** starting part B.

Task 2 – part B requires candidates to complete a data dictionary. They must submit their evidence for this part of the task **before** starting part C.

Task 2 – part C is a separate section. This ensures that candidates do not access part B and change their responses. A Microsoft Access file (AuctionDatabase) is provided for candidates to use in part B. If your centre uses a different database management system, you can create the relational database for part B using the CSV files provided:

If using the CSV files you should set up all tables, fields and validation shown in the data dictionaries below. Referential integrity should also be enforced.

If your centre uses an SQL server, you can create the database using the text files provided. These files contain SQL create and insert statements for each table. If you use the text files, you must add validation (shown in the data dictionaries below), appropriate for your version of SQL. Referential integrity should also be enforced.

Entity: Seller					
Attribute name	Key	Type	Size	Required	Validation
sellerName		text	50	Y	
sellerAddress		text	255	Y	
sellerID	PK	text	4	Y	
email		text	100	N	
telephoneNumber		text	11	Y	Length = 11

Entity: Property					
Attribute name	Key	Type	Size	Required	Validation
houseNumber		number		Y	
street		text	75	Y	
city		text	9	Y	Restricted choice: Aberdeen, Dundee, Edinburgh, Glasgow, Inverness, Perth, Stirling
postcode		text	8	Y	
propertyRef	PK	text	15	Y	
sellerID	FK	text	4	Y	
numberOfBedrooms		number		Y	
estimatedValue		number		Y	Range: >= 0 AND <= 300000000
askingPrice		number		Y	Range: >= 0 AND <= 300000000

Task 3 – part A requires candidates to identify end-user and functional requirements. They must submit their evidence to you **before** you issue part B.

Task 3 – part B is a separate section. This ensures that candidates do not access part A and change their responses.

A file named ‘Customised Cakes’ is provided. This contains the CSS, HTML and media files candidates need to complete this task. These files should remain in the folder and should not be renamed.

Candidates **do not** need to print the completed web pages in colour.

Instructions for candidates

This assessment applies to the assignment for National 5 Computing Science.

This assignment has 40 marks out of a total of 120 marks available for the course assessment.

It assesses the following skills, knowledge and understanding:

- ◆ applying aspects of computational thinking across a range of contexts
- ◆ analysing problems within computing science across a range of contemporary contexts
- ◆ designing, implementing, testing and evaluating digital solutions (including computer programs) to problems across a range of contemporary contexts
- ◆ demonstrating skills in computer programming
- ◆ applying computing science concepts and techniques to create solutions across a range of contexts

Your teacher or lecturer will let you know if there are any specific conditions for doing this assessment.

In this assessment, you have to complete **two** short practical tasks. You may complete the tasks in any order.

You must complete task 1 (software design and development) and **either** task 2 (database design and development) **or** task 3 (web design and development).

Advice on how to plan your time

You have 6 hours to complete the assignment. Marks are allocated as follows:

- | | | |
|--|----------|----------------|
| ◆ Task 1 – software design and development | 25 marks | (63% of total) |
|--|----------|----------------|

AND EITHER

- | | | |
|--|----------|----------------|
| ◆ Task 2 – database design and development | 15 marks | (37% of total) |
|--|----------|----------------|

OR

- | | | |
|-------------------------------------|----------|----------------|
| Task 3 – web design and development | 15 marks | (37% of total) |
|-------------------------------------|----------|----------------|

You can use this split as a guide when planning your time for each of the two tasks.

Advice on gathering evidence

As you complete each task, you must gather evidence as instructed in each task.

Your evidence, especially code, must be clear and legible. This is particularly important when you paste screenshots into a document.

Use the evidence checklist provided to make sure you submit everything necessary at the end of the assignment. Ensure your name and candidate number is included on all your evidence.

Evidence may take the form of printouts of code/screenshots/typed answers, hand-written answers or drawings of diagrams/designs.

Advice on assistance

This is an open-book assessment. This means that you can use:

- ◆ any classroom resource as a form of reference (for example programming manuals, class notes, and textbooks) – these may be online resources
- ◆ any files you have previously created throughout the course

The tasks are designed so you can complete them independently, without any support from your teacher or lecturer. This means that you:

- ◆ cannot ask how to complete any of the tasks
- ◆ cannot access any assignment files outside the classroom

Computing Science assessment task: evidence checklist

You should complete the checklist for task 1 and either task 2 or task 3.

Task 1	Evidence	
Part A		
1a	Completed task sheet showing the expanded design for step 3	☐
Part B		
1b	Printout of your program code	☐
1c (i)	Printout evidence of the test run showing inputs and outputs	☐
1c (ii)	Completed task sheet showing the completed test table	☐
1d	Completed task sheet with your program evaluation	☐

Task 2	Evidence	
Part A		
2a	Completed task sheet showing the analysis of inputs table	☐
Part B		
2b	Completed task sheet showing the completed data dictionary	☐
Part C		
2c (i)	Printout of SQL statement to change the asking price	☐
	Printout of the updated Property table	☐
2c (ii)	Printout of SQL statement to add a new seller's details	☐
	Printout of the updated Seller table	☐
2c (iii)	Printout of SQL statement to display a list of sellers' details	☐
	Printout of the output of the query	☐
2d	Completed task sheet with two reasons for wrong output	☐
2e	Completed task sheet with your evaluation of the database	☐

Task 3	Evidence	
Part A		
3a	Completed task sheet stating one user requirement and two functional requirements for the website	☐
Part B		
3b	Completed task sheet showing the wireframe design for the 'Recommend a Friend' page	☐
3c, 3d, 3e	Printouts of HTML and CSS code: <ul style="list-style-type: none">◆ home.html◆ recommend.html◆ styles.css	☐
3f	Completed task sheet with your evaluation of the finished website	☐

Please follow the steps below before handing your evidence to your teacher or lecturer:

- ◆ Check you have completed all parts of task 1 and one of **either** task 2 or 3.
- ◆ Label any printouts/screenshots with the task number (for example 1a, 2a).
- ◆ Clearly display your name and candidate number on each printout.

Task 1: software design and development (part A)

AutoFeed has designed an automatic dog food dispenser. The product is designed to help dog owners prevent overfeeding. The food dispenser has five containers. Each container releases food, one at a time, at regular intervals throughout the day. The containers can hold a maximum of 200g of food.

AutoFeed wants an app to allow users to enter information about the size of their dog and the weight of food they will give them. The app uses this data to show if the weight of food is suitable for the size of dog.

AutoFeed research recommends the following weight of food (per day), depending on the size of dog.

Size of dog	Recommended weight of food per day
Small	From 110g to 140g
Medium	From 330g to 440g
Large	From 690g to 900g

After analysing the problem, the following inputs, processes and outputs have been identified.

Inputs

- ♦ the weight of food in each container
- ♦ the size of dog (small, medium or large)

Processes

- ♦ calculate the total weight of food in the five containers
- ♦ store a message that states if the total weight of food is within the recommended range
- ♦ calculate the average weight of food in the five containers
- ♦ round the average weight of food to one decimal place

Outputs

- ♦ the weight of food in each container
- ♦ the average weight of food in the five containers
- ♦ the rounded average weight of food
- ♦ the stored message stating if the total weight of food is within the recommended range

Assumptions

- ♦ the user can enter a weight using whole numbers or real numbers
- ♦ the user does not need to fill all five containers
- ♦ the user will not refill any of the containers until all containers are empty

Program design (pseudocode)

Main steps

- 1 Enter valid weight of food in each container and calculate total weight
- 2 Enter size of dog
- 3 Store a message that states if the total weight of food is within the recommended range
- 4 Calculate average weight
- 5 Display output messages

Refinements

- 1.1 totalWeight = 0
- 1.2 start fixed loop 5 times
- 1.3 enter the foodWeight
- 1.4 while the foodWeight < 0 or foodWeight > 200
- 1.5 display "Invalid, a single container can only hold up to 200g"
- 1.6 re-enter the foodWeight
- 1.7 end while
- 1.8 totalWeight = totalWeight + foodWeight
- 1.9 end fixed loop

- 2.1 display "Please enter the size of your dog: small, medium or large"
- 2.2 enter size of dog

- 4.1 averageWeight = totalWeight / 5
- 4.2 round averageWeight to 1 decimal place

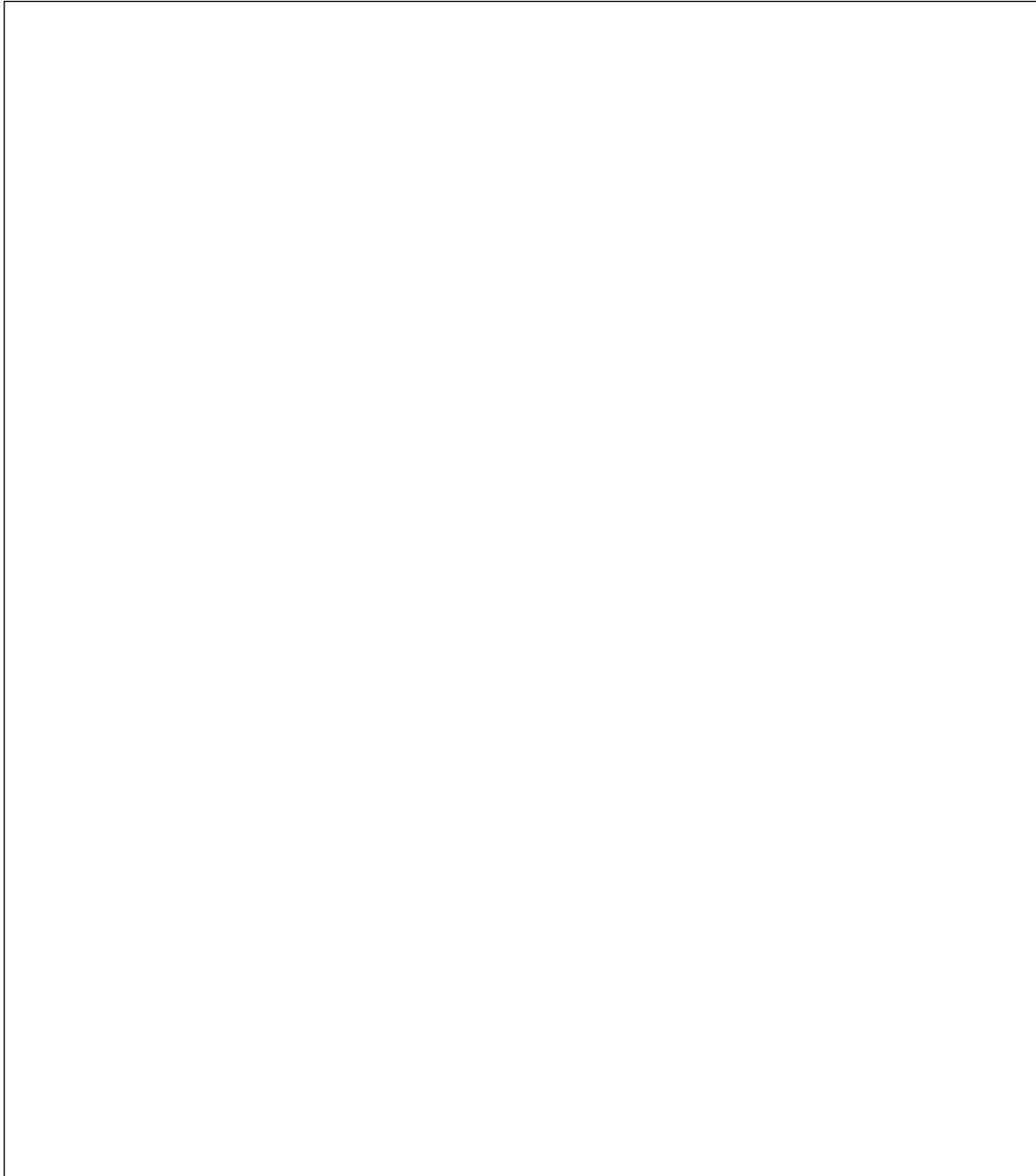
- 5.1 start fixed loop 5 times
- 5.2 display next foodWeight
- 5.3 end fixed loop
- 5.4 display total weight message
- 5.5 display average weight message
- 5.6 display recommendation message

1a Step 3 of the program design is incomplete:

- 3 Store a message that states if the total weight of food is within the recommended range

Using the information provided in the program analysis, expand the design to show how this process could be carried out. You can use a flowchart, structure diagram or pseudocode design.

(3 marks)



- ♦ Check your answers carefully, as you cannot return to part A after you hand it in.
- ♦ When you are ready, hand part A to your teacher or lecturer and collect part B.

Candidate name_____ Candidate number_____

Task 1: software design and development (part B)

Program design (completed pseudocode)

Main steps

- 1 Enter valid weight of food in each container and calculate total weight
- 2 Enter size of dog
- 3 Store a message that states if the total weight of food is within the recommended range
- 4 Calculate average weight
- 5 Display output messages

Refinements

- 1.1 totalWeight = 0
- 1.2 start fixed loop 5 times
- 1.3 enter the foodWeight
- 1.4 while the foodWeight < 0 or foodWeight > 200
- 1.5 display "Invalid, a single container can only hold up to 200g"
- 1.6 re-enter the foodWeight
- 1.7 end while
- 1.8 totalWeight = totalWeight + foodWeight
- 1.9 end fixed loop

- 2.1 display "Please enter the size of your dog: small, medium or large"
- 2.2 enter size of dog

- 3.1 if dog size = small and totalWeight is from 110 to 140 then
- 3.2 store message "This weight of food is suitable for your small dog."
- 3.3 else
- 3.4 if dog size = medium and totalWeight is from 330 to 440 then
- 3.5 store message "This weight of food is suitable for your medium dog."
- 3.6 else
- 3.7 if dog size = large and totalWeight is from 690 to 900 then
- 3.8 store message "This weight of food is suitable for your large dog."
- 3.9 else
- 3.10 store message "This weight of food is not recommended for the size of dog"
- 3.11 end if
- 3.12 end if
- 3.13 end if

- 4.1 averageWeight = totalWeight / 5
- 4.2 round averageWeight to 1 decimal place

- 5.1 start fixed loop 5 times
- 5.2 display next foodWeight
- 5.3 end fixed loop
- 5.4 display total weight message
- 5.5 display average weight message
- 5.6 display recommendation message

- 1b Using the program analysis and the design, implement the program in a language of your choice.

Make sure the program matches the pseudocode provided on page 14.

(15 marks)

Print evidence of your program code.

- 1c (i) Test your program to make sure it produces the expected output.

Use the following test data to check that the message “This weight of food is suitable for your medium dog” is displayed:

Weight 1: 134.23

Weight 2: 74.99

Weight 3: 25.31

Weight 4: 112.33

Weight 5: 53.78

Size of dog: medium

Print evidence of the test run showing all inputs and the message displayed.

(1 mark)

- (ii) Additional test data is required to check that the correct output messages are displayed.

Complete the test table below to show the expected results for Test 1 and appropriate inputs for Test 2.

(2 marks)

	Type of test	Expected results
Test 1	Weight 1: 30 Weight 2: 50 Weight 3: 45 Weight 4: 150 Weight 5: 70 Size of dog: large	
Test 2	Weight 1: Weight 2: Weight 3: Weight 4: Weight 5: Size of dog:	"This weight of food is suitable for your small dog."

Candidate name_____ Candidate number_____

1d With reference to your code, evaluate your program by commenting on the following:

Efficiency of your program code	(2 marks)
Robustness of your completed program	(1 mark)
Readability of your code	(1 mark)

Candidate name_____ Candidate number_____

Task 2: database design and development (part A)

ScotAuction sells property in Aberdeen, Dundee, Edinburgh, Glasgow, Inverness, Perth and Stirling.

To register a property for sale, ScotAuction requests the following details from sellers: name, email address, telephone number and home address. ScotAuction gives each seller a unique ID and will contact the seller if an offer is made.

Every property registered for sale must have the following information recorded: house number, street, city, postcode, asking price and estimated property value. ScotAuction generates a unique reference code for all properties. Sellers should also state the number of bedrooms in the property.

A seller can have multiple properties for sale.

A property cannot be listed for sale without the seller's details being recorded first.

2a ScotAuction wants to create a database to store the seller and property details.

Complete the property details in the analysis of inputs table below.

(2 marks)

Seller details:	Property details:
Seller ID Seller name Seller address Email address Telephone number	

- ♦ Check your answers carefully, as you cannot return to part A after you hand it in.
- ♦ When you are ready, hand part A to your teacher or lecturer and collect part B.

Candidate name_____ Candidate number_____

Task 2: database design and development (part B)

- 2b Complete the data dictionary below, by identifying the primary and foreign keys in the relational database.

(2 marks)

Entity: Seller					
Attribute name	Key	Type	Size	Required	Validation
sellerName		text	50	Y	
sellerAddress		text	255	Y	
sellerID		text	4	Y	
email		text	100	N	
telephoneNumber		text	11	Y	Length = 11

Entity: Property					
Attribute name	Key	Type	Size	Required	Validation
houseNumber		number		Y	
street		text	75	Y	
city		text	9	Y	Restricted choice: Aberdeen, Dundee, Edinburgh, Glasgow, Inverness, Perth, Stirling
postcode		text	8	Y	
propertyRef		text	15	Y	
sellerID		text	4	Y	
numberOfBedrooms		number		Y	
estimatedValue		number		Y	Range: >= 0 AND <= 300000000
askingPrice		number		Y	Range: >= 0 AND <= 300000000

- ◆ Check your answers carefully, as you cannot return to part B after you hand it in.
- ◆ When you are ready, hand part B to your teacher or lecturer and collect part C.

Candidate name_____ Candidate number_____

Task 2: database design and development (part C)

2c Your teacher or lecturer will give you a database file containing two linked tables.

- (i) The asking price for propertyRef DUN-101 has been recorded incorrectly. This should be changed from £105500 to £112000.

Implement an SQL statement that will make the required change to the asking price.

(2 marks)

Print evidence of the SQL statement and the Property table, clearly showing that the change has been implemented.

- (ii) Implement an SQL statement that will add the following details of a new seller to the database.

sellerID:	1502
sellerName:	Eve Grace
sellerAddress:	128 Cameron Drive Edinburgh EH4 5DS
email:	EveGrace@yehoo.net
telephoneNumber:	0131 279100

(2 marks)

Print evidence of the SQL statement and the Seller table, clearly showing that the change has been implemented.

- (iii) ScotAuction is running a workshop to give sellers advice on how to achieve a higher sale price. Due to the limited spaces available, ScotAuction is only inviting selected sellers to attend.

Implement an SQL statement that will display the seller's email address and telephone number and the property's postcode for properties that have three bedrooms and an asking price of less than £150000.

(4 marks)

Print evidence of the SQL statement and the output.

- 2d ScotAuction would like a list of seller IDs and asking prices for properties in Glasgow. The list should be sorted showing the lowest price first.

The following incorrect SQL statement is written:

```
SELECT Seller.sellerID, price
FROM Seller, Property
WHERE town = "Glasgow" AND Seller.sellerID = Property.sellerID
ORDER BY askingPrice ASC;
```

Test this SQL statement.

State two reasons why this SQL statement produces the wrong output.

(2 marks)

Reason 1
Reason 2

- 2e The initial analysis identified the following functional requirements for the database. It should:

- ◆ allow ScotAuction to store a seller's email address and telephone number
- ◆ allow sellers to state an estimated value of their property
- ◆ record key details about each property, including the address and the number of bedrooms
- ◆ limit the value and location of the properties that can be entered

Use the above analysis to evaluate the database in terms of fitness for purpose.

(1 mark)

--

Candidate name_____ Candidate number_____

Task 3: web design and development (part A)

Customised Cakes makes and sells custom cakes for family events and celebrations. This includes birthday cakes, wedding cakes as well as cakes for other events.

It wants a website containing the following elements:

- ◆ the title 'Customised Cakes'
- ◆ a short statement about the shop's promise to customers
- ◆ separate pages for each category of cake: birthday cakes, wedding cakes and other special occasions
- ◆ a 'Recommend a Friend' page that gives details on how to register for rewards for you and your friend
- ◆ at least one photograph on each page showing cake designs
- ◆ a video on the home page showing a cake being iced
- ◆ an external link on the 'Recommend a Friend' page to 'The Great British Bake Off' website

3a State **one** end-user and **two** functional requirements for the website.

End-user requirement	(1 mark)
Functional requirement 1	(1 mark)
Functional requirement 2	(1 mark)

Candidate name_____ Candidate number_____

- ◆ Check your answers carefully, as you cannot return to part A after you hand it in.
- ◆ When you are ready, hand part A to your teacher or lecturer and collect part B.

Task 3: web design and development (part B)

3b Your teacher or lecturer will give you a copy of the unfinished website.

Open the website and look carefully:

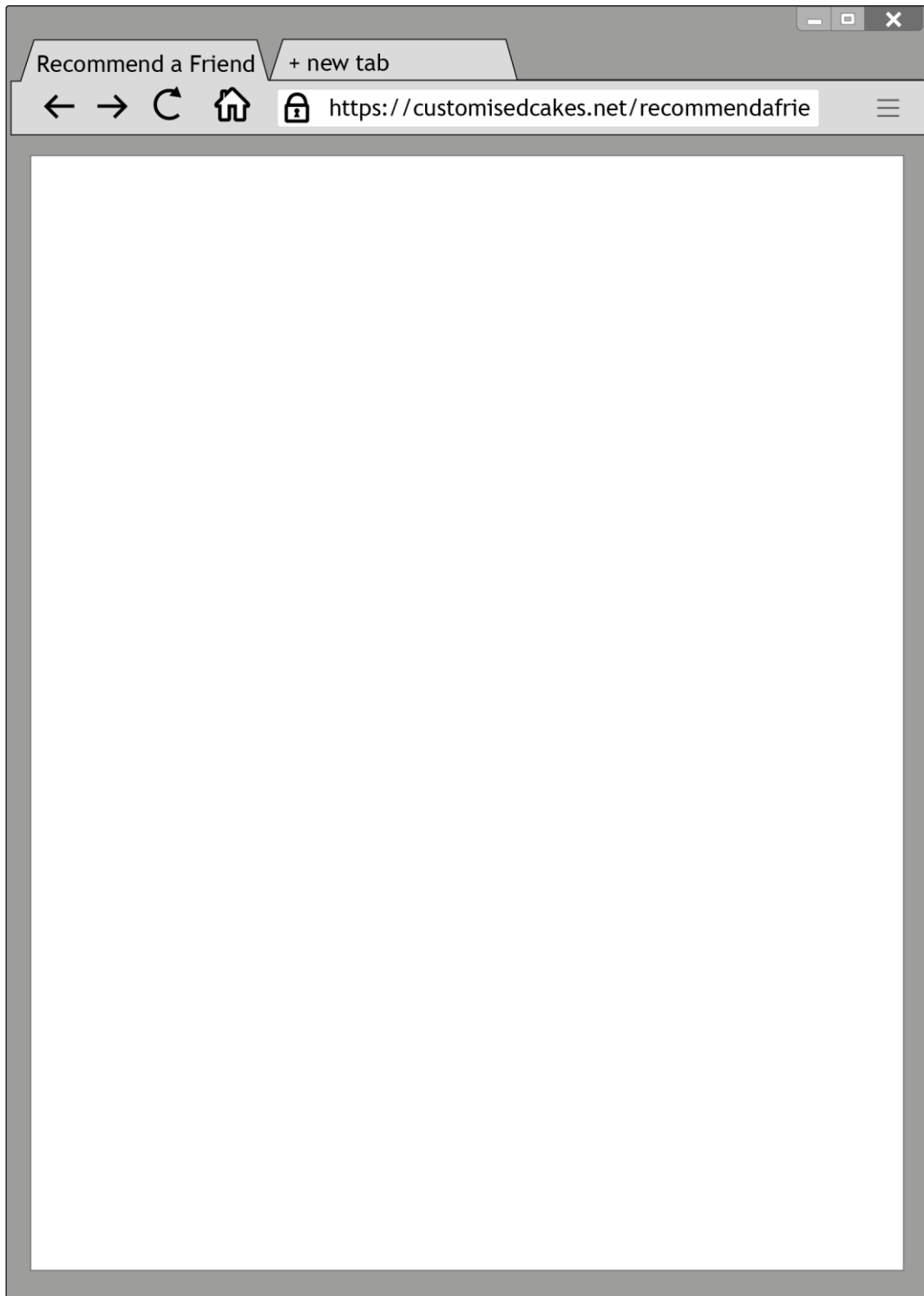
- ◆ at the layout of each page
- ◆ at the content of each page
- ◆ at the navigation within the website
- ◆ for any text and graphics that appear on every page

The 'Recommend a Friend' page currently has no content. When complete, it should be consistent with the other pages and include the following additional content:

- ◆ a message stating that for every friend you recommend who places an order, you will receive a free slice of cake
- ◆ a message stating that both you and your friend will receive 20% off your next order
- ◆ a message stating that most orders are delivered within three days
- ◆ a message telling the user that they can take part in the 'Recommend a Friend' scheme by emailing their friend's name and email address to ReferAFriend@CustomisedCakes.net
- ◆ an image (300 x 240 pixels) of a slice of cake
- ◆ an external link to 'The Great British Bake Off' website <https://www.thegreatbritishbakeoff.co.uk>
- ◆ a hyperlink back to the home page

Complete the wireframe on the following page, showing the website layout and the content for the 'Recommend a Friend' page.

(2 marks)



Candidate name_____ Candidate number_____

3c Open the recommend.html and styles.css files so you can edit them.

Implement your design of the 'Recommend a Friend' web page using HTML, including all of the content and any hyperlinks required.

An image of a slice of cake ('cakeSlice.jpg') is provided in the files.

(4 marks)

3d Customsied Cakes wants to add the video file 'cakeVideo.mp4' to the homepage.

Edit the home.html file to implement this change.

(1 mark)

Print evidence of your code in the edited home.html file.

3e The table below shows feedback received during testing and the required changes:

Feedback received	Required changes
"The website would look better if the coloured sections on each page were the same colour."	Change the coloured sections to DeepSkyBlue (#00BFFF).
"The company motto 'Beautiful, hand-crafted custom cakes for any occasion' needs to look more professional."	Change the font size to 22 and the font to Arial.
"The link to 'The Great British Bake Off' website does not stand out enough."	Change the style of the text in the link https://www.thegreatbritishbakeoff.co.uk to colour OrangeRed (#FF4000) with a font size of 16.

Edit the recommend.html and styles.css files to implement these changes.

(3 marks)

Print evidence of your code from these edited files:

- ♦ recommend.html
- ♦ styles.css

3f Initial analysis identified the following functional requirements for the website. It should display:

- ◆ the name of the shop and the shop's customer promise
- ◆ at least one image of each type of cake
- ◆ details of the 'Recommend a Friend' promotion
- ◆ a video of a cake being iced
- ◆ a link for each page of the site and a link to 'The Great British Bake Off' website

Use the above analysis to evaluate your finished website in terms of fitness for purpose.

(2 marks)

Candidate name_____ Candidate number_____

Copyright acknowledgements

Electronic files:

Timmary/Shutterstock.com
archideaphoto/Shutterstock.com
Unal Ozmen/Shutterstock.com
MaraZe/Shutterstock.com
Ruth Black/Shutterstock.com
October22/Shutterstock.com
Ahanov Michael/Shutterstock.com
muhammed yasin irik/Shutterstock.com
Mariko151825/Shutterstock.com

Administrative information

Published: January 2022 (version 1.0)

History of changes

Version	Description of change	Date

Security and confidentiality

This document can be used by SQA approved centres for the assessment of National Courses and not for any other purpose.

This document may only be downloaded from SQA's designated secure website by authorised personnel.

© Scottish Qualifications Authority 2022