National 5
Coursework
Assessment Task

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 2024-25 only.

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

This edition: January 2025 (version 1.0)

Contents

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 2024–25.

The task is valid for 2024–25 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 e-mail 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

You can use any integrated development environments (IDE) that enables candidates to generate evidence — this includes online IDEs. However, the IDE must have a facility that prevents candidates accessing their files and tasks outside the supervised 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**.

You can provide candidates with templates, however these templates must only contain general starter code used in learning and teaching (for example, a web page that contains the HTML, title and body elements) — templates must not be tailored to this year's task.

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 handwritten or created 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.

If evidence is handwritten, candidates must use a blue or black pen.

When packaging, ensure that:

- each assignment is accompanied by an SQA A4 flyleaf
- all completed task sheets and additional evidence of screenshots or printouts from development environments are included and ordered in line with the task
- where possible, sheets that do not contain candidate evidence are removed
- candidates' SCNs are on every page
- the flyleaf and candidate evidence are stapled together in the top left corner

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 as 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 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: 2024-25

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:

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

Task 1 — part A requires candidates to identify processes and complete a design. They **must** submit their evidence to you before you issue part B.

Task 1 — part B is a separate section. This ensures that candidates do not access part A and change their responses. Candidates must still have access to the program description during part B.

Task 2 — part A requires candidates to complete an analysis and data dictionary. They must submit their evidence to you before you issue part B.

Task 2 — part B is a separate section. This ensures that candidates do not access part A and change their responses.

A Microsoft Access file (plantDB.accdb) is provided for candidates to use in part B. If your centre uses a different database management system, you can create the relational database using the CSV files or the text 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.

The text 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: Climate

Attribute name	Key	Type	Size	Required	Validation
climateRef	PK	number		Y	
climateType		text	50	Y	
temperatureRange		text	20	Y	
humidityRange		text	20	Y	

Entity: Plant

Attribute name	Key	Type	Size	Required	Validation
plantID	PK	number		Y	
climateRef	FK	number		Y	Existing climateRef from Climate table
plantName		text	100	Y	
scientificName		text	50	Y	
ediblePart		text	50	Y	
soilType		text	5	N	Restricted choice: Clay, Loam, Peat or Sandy

Task 3 — A folder titled 'Web files' is provided. This contains the HTML and media files candidates need to complete this task. These files must not be renamed and they must remain in the folders provided. However, the case of suffixes may be changed if the environment you work in requires them to be lower or upper case.

Candidates **do not** need to print 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 must complete task 1 (software design and development) and **either** task 2 (database design and development) **or** task 3 (web design and development).

You may complete the tasks in any order.

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.

Your evidence, especially code, must be clear and legible. This is particularly important when you paste screenshots into a document. You can print code from the software environment or copy and paste this into other packages such as notepad or Word.

Use the evidence checklist provided to make sure you submit everything necessary at the end of the assignment. Make sure you include your candidate number on all your evidence.

Evidence may take the form of printouts of code, screenshots, typed answers, handwritten answers or drawings of diagrams and designs.

You must use a blue or black pen for any handwritten answers.

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 — software design and development

Task	Evidence	Tick
1a	Completed task sheet showing additional processes	
1b	Completed task sheet showing your design	
1c	Completed task sheet showing expected output	
1d	Printout of your program code and program output	
1e	Completed task sheet with your evaluation	

Task 2 — database design and development

Task	Evidence	Tick
2a	Completed task sheet showing functional requirements	
2b	Completed task sheet showing the analysis of the inputs	
2c	Completed task sheet showing the completed data dictionary	
2d (i)	Printout of SQL statement to add new climate type Printout of the updated table	
2d (ii)	Printout of SQL statement	
2d (iii)	Printout of SQL statement Printout of the output	
2e	Completed task sheet stating reasons why the SQL statement does not produce the expected output	

Task 3 — web design and development

Task	Evidence	Tick
3a	Completed task sheet with two functional requirements	
3b	Printout of edited code for amazingMenzies.html	
3c	Completed task sheet with description of test that could be performed on the web page	
3d	Completed task sheet with your explanation of how a JavaScript event could make the web page interactive	
3e	Completed task sheet with your website structure diagram	

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

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

Task 1: software design and development

Shake-Shake creates fruit-based drinks. They need a computer program that will tell their customers if selected fruits would make a better milkshake, smoothie or fruit juice.

Program description

Shake-Shake wants a computer program that allows users to enter the names of up to 6 fruits.

The user enters the name of each fruit they want in their drink. The program will validate each fruit name by checking it is at least 4 characters long.

The program then adds a mystery fruit, which it randomly selects from these options:

- apple
- banana
- blueberry
- kiwi
- mango
- orange
- peach
- pineapple
- raspberry
- strawberry

Once the user enters all their fruit options, the program adds the mystery fruit and uses the total number of fruits to determine whether it recommends a milkshake, smoothie or fruit juice.

Shake-Shake recommends that less than 3 fruits should be a milkshake, 3 or 4 fruits should be a smoothie, and more than 4 fruits should be a fruit juice.

The program will output a summary of the user's fruit choice, including the mystery fruit, and a message advising what type of drink to make.

Task 1: software design and development (part A)

1a Complete the table by identifying two more processes.

(2 marks)

Input
Name of each fruit
Processes
Check no more than 6 fruits entered Select mystery fruit option Calculate total number of fruits
Outputs
User's fruit choices Mystery fruit Message recommending type of drink (milkshake, smoothie or fruit juice)

Candidate number_____

1b The program will validate a fruit name by checking it is at least 4 characters long.

Design how this could be implemented. You can use a flowchart, structure diagram or pseudocode design.

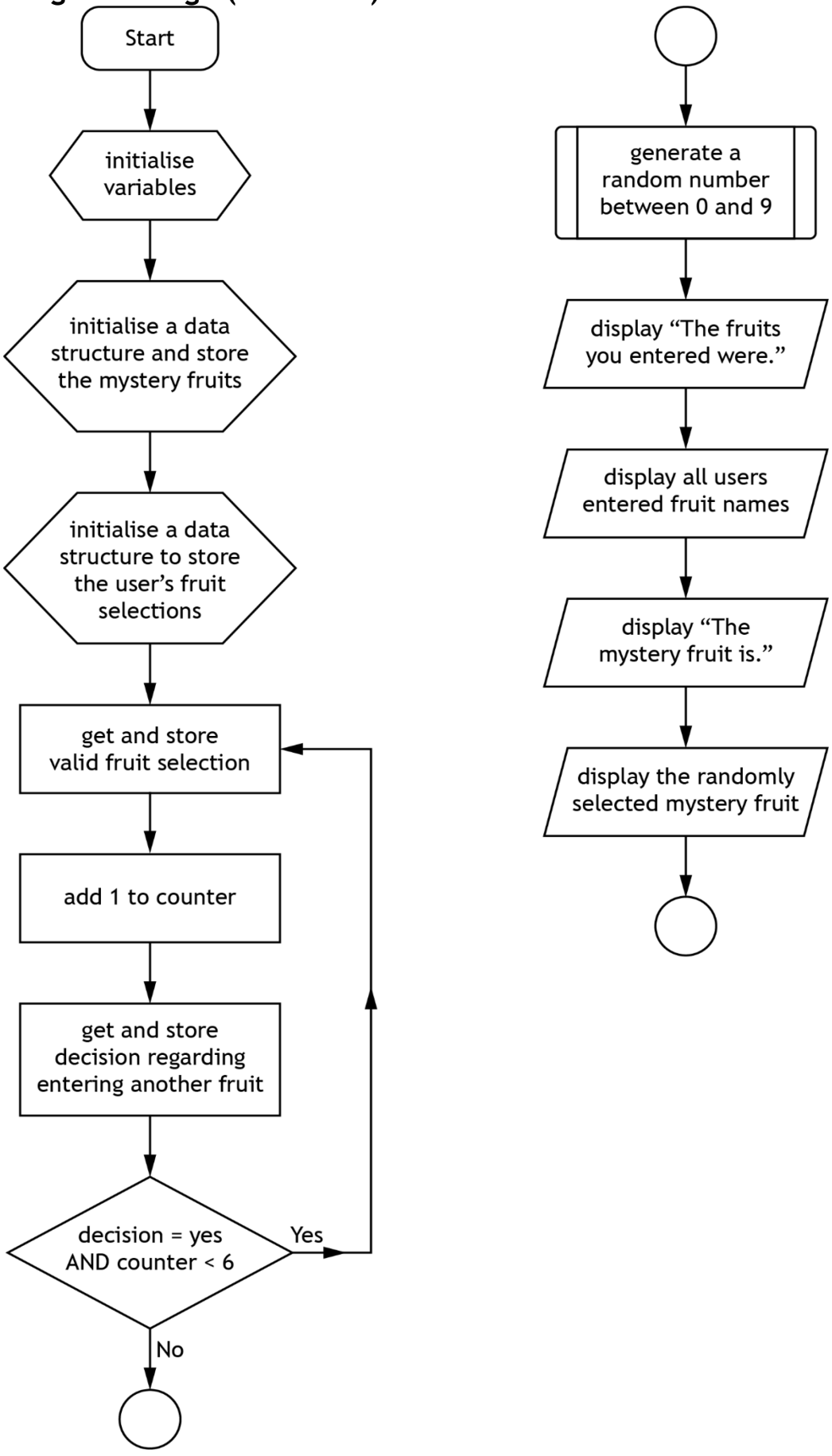
(3 marks)

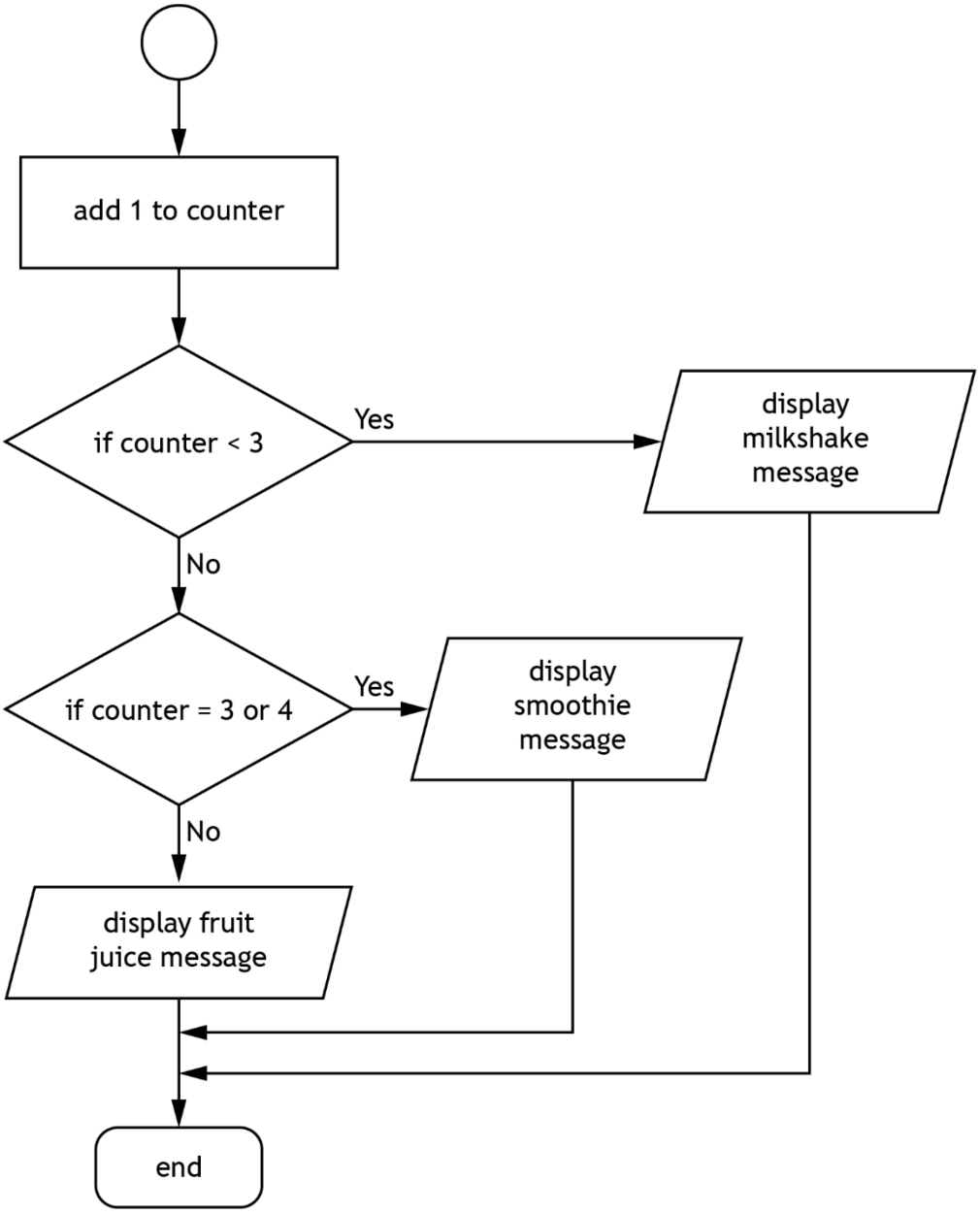
- Check your answers carefully, as you cannot return to part A after you hand it in.
- When you are ready, hand part A to your teacher or lecturer and collect part B.

Candidate number_____

Task 1: software design and development (part B)

Program design (flowchart)




add 1 to counter
if counter < 3
Yes
display milkshake message
No
if counter = 3 or 4
Yes
display smoothie message
No
display fruit juice message
end

1c Using the program design, complete the expected output in the test table below. Use mango as the mystery fruit.

Type of test	User input	Expected output
Normal	Selection 1: **lemon** Selection 2: **pear** Selection 3: **peach** Selection 4: **orange**	

(2 marks)

1d Using the program description and the design, implement the program in a language of your choice.

Make sure the program matches the design given.

Run your completed program entering two fruits of your choice.

(15 marks)

Print evidence of your program code and your program output.

Candidate number_____

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

Use of efficient programming constructs	**(1 mark)**
Robustness of your program	**(1 mark)**
Readability of your code	**(1 mark)**

Candidate number_____

Task 2: database design and development (part A)

An environmental group wants to encourage people to grow their own fruit and vegetables. The group stores details on edible plants and the climates they grow in.

The group plans to create a database to store details about plants and climates. They have been discussing the end-user requirements of the new database.

They concluded that:

- They need to know the details of climates, including their humidity and temperature ranges.
- They already have a lot of information about different plants, for example carrot, lettuce and watermelon. They would like to add to the database when they have information about other plants.
- They want to know which plants grow best in a particular climate.
- They need to be able to amend their data when they receive new research.

2a Using the information above, create two functional requirements of the database.

(2 marks)

Functional requirement 1
Functional requirement 2

Candidate number_____

The information on climates and plants will be stored in a database.

Each climate will be assigned a unique identifier and a climate type (temperate, tropical, mediterranean or arid). The associated edible plants each have a unique identifier. Information about the plant's name, its scientific name, the edible part of the plant and the type of soil required to grow it (clay, loam, peat or sandy) will be stored. The temperature and humidity for each climate will be stored.

2b Complete the analysis of the inputs shown below:

(2 marks)

Climate	**Plant**
	Plant ID Plant name Scientific name Edible part Type of soil

Candidate number_____

2c Complete the data dictionary for the 'Plant' entity by:

- completing type, size, required and validation for the soilType attribute
- identifying the foreign key

(2 marks)

Entity: Plant

Attribute name	Key	Type	Size	Required	Validation
plantID	PK	number		Y	
climateRef		number		Y	
plantName		text	100	Y	
scientificName		text	50	Y	
ediblePart		text	50	Y	
soilType					

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

Candidate number_____

Task 2: database design and development (part B)

2d Your teacher or lecturer will give you a completed database that includes data on plants and climates.

i) A new climate is to be added to the database. Implement an SQL statement that will add the following climate:

climateRef: 105
climate type: Temperate Oceanic
temperature range: 15-20
humidity range: 70-80

Print evidence of your SQL statement and evidence clearly showing that the change has been implemented.

(1 mark)

ii) ClimateRef '103' is the best climate for all plants that have an edible fruit.

Implement an SQL statement that will change the recommended climateRef to '103' where the edible part of the plant is 'Fruit'.

Print evidence of your SQL statement.

(2 marks)

iii) The group wants to promote growing edible leaves.

Implement an SQL statement that will display the plant name, climate type, edible part, soil type and temperature of all plants that grow edible leaves. Show the results in order of temperature, with the highest temperature first.

Print evidence of your SQL statement and the output from the query after it has been implemented.

(4 marks)

2e The following SQL statement is written to find the climateRef, temperature range, plant name and edible part of all plants that grow in loam soil and a Mediterranean climate.

```
SELECT Climate.climateRef, temperatureRange, plantName,
ediblePart
FROM Climate
WHERE soilType = "Loam"
AND climateType = "Mediterranean";
```

Test this SQL statement.

State two reasons why this SQL statement does not produce the expected output.

(2 marks)

Reason 1
Reason 2

Candidate number_____

Task 3: web design and development

The 'Amazing Menzies' invents new products. They want a web page to showcase their ideas. The web page needs to include:

- the name of each invention with a description:
  - silent crisp packet
  - self-charging smartphone
  - noiseless Velcro
  - water purification straw
- an image of each invention (width 300px by height 200px)
- an external link to the Scottish Enterprise website: https://www.scottish-enterprise.com
- a video demonstrating the silent crisp packet

3a Create two functional requirements for this web page.

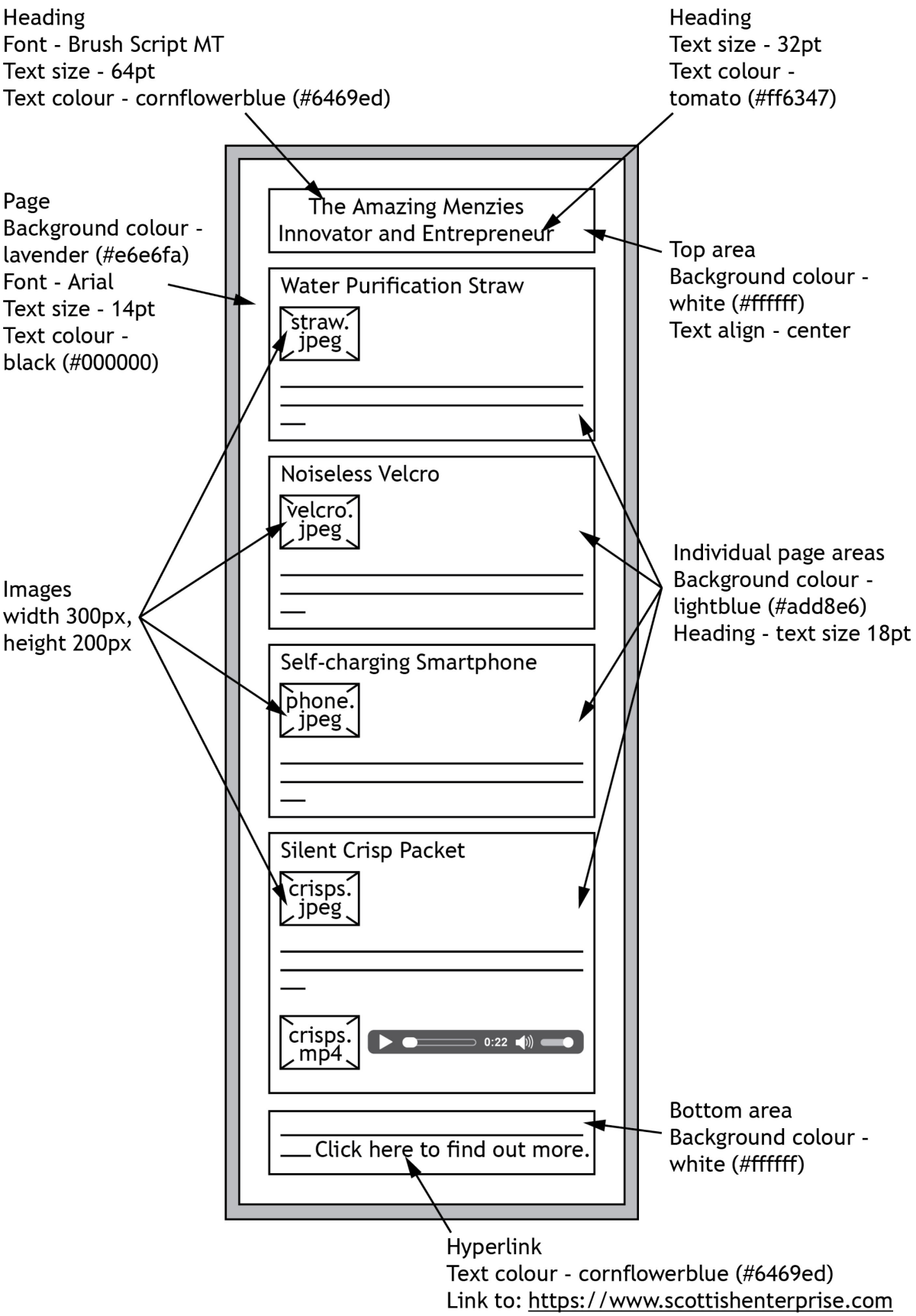
(2 marks)

Functional requirement 1
Functional requirement 2

Candidate number_____

When designing the web page, a wireframe and a low-fidelity prototype of the page are produced.

The wireframe design of the page (annotated with required styles for each coloured area of the page) is shown below:



The low-fidelity prototype for the page is shown below:

3b Your teacher or lecturer will give you the partially complete HTML file:

- amazingMenzies.html

and the following media files:

- crisps.jpeg
- phone.jpeg
- straw.jpeg
- velcro.jpeg
- crisps.mp4

Implement the design using HTML and internal CSS.

(8 marks)

Print evidence of the following:

- the completed 'amazingMenzies.html' file
- the web page as viewed in a web browser

3c Describe one test that you have performed on this web page.

(1 mark)

3d Explain how a JavaScript event could be used to add interactivity to this web page.

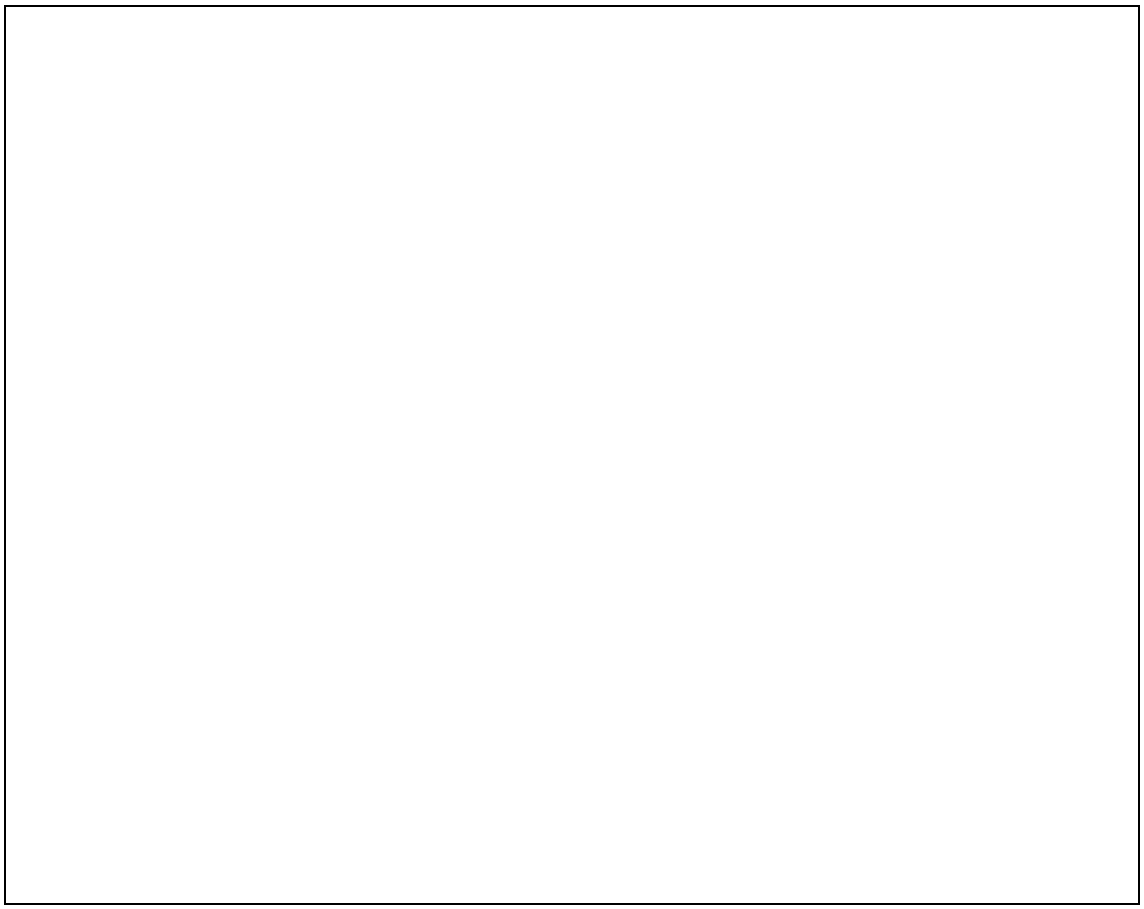
(2 marks)

Candidate number_____

3e Amazing Menzies would like the design to be split over multiple web pages.

Draw a navigational structure diagram to show how the web page could be split into different pages, showing all links.

(2 marks)



Candidate number _____

Copyright Acknowledgements

Task 3 - Still of video of crisp packet – Sved Oliver/Shutterstock.com

Electronic Files:

Video of crisp packet – Sved Oliver/Shutterstock.com

Administrative information

Published: January 2025 (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.