



# Higher 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 2023-24 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 Higher 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 Higher Computing Science for the academic session 2023-24.

The task is valid for 2023-24 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

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**:

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

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 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](mailto: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: 2023-24

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 analyse a software problem. 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 problem description during part B.

**Task 1 – part C** is a separate section. This ensures that candidates do not access part B and change their responses.

A CSV file (companies.csv) is provided for candidates to use in part B and part C. You must not convert the CSV file into a different format. Candidates are assessed on their ability to implement a solution to the given task, using the specific file type provided.

**Task 2 – part A** requires candidates to analyse a database. 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 B and change their responses.

A Microsoft Access file (NSA.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 for using the CSV files or the text files provided.

If using the CSV files, you should set up all tables, fields and validation as shown below.

The text files contain SQL create and insert statements for each table.

The CSV files contain the data for each table. The text files contain SQL create and insert statements for each table. In both cases, you will have to add primary keys and foreign keys as specified below.

## NSA database

Event	Race	Result	Swimmer	Team
<u>eventID</u>	<u>raceNumber</u>	<u>raceNumber*</u>	<u>swimmerID</u>	<u>teamRef</u>
eventDate	raceCategory	position	initial	teamName
city	stroke	lane	surname	headCoach
venue	distance	<u>swimmerID*</u>	swimCategory	assistantCoach
	eventID*	raceTime	teamRef*	

**Task 2e** – requires candidates to test an SQL statement. You must provide this to candidates as part of the database. The SQL statement is already included in the MS Access file. If you use a different database management system, you should use the supplied text file (Query for 2e.txt) to add it to the database you provide to candidates.

**Task 3 – part A** requires candidates to design a multi-level navigation structure for a website. 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 folder named 'Web files' is provided. This contains the CSS, 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 Higher 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) |
| <b>AND EITHER</b>                          |          |                |
| ◆ Task 2 – database design and development | 15 marks | (37% of total) |
| <b>OR</b>                                  |          |                |
| ◆ 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. 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. 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 and 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 – software design and development

Evidence		Tick
1a	Completed task sheet identifying the missing functional requirements	
1b	Completed task sheet showing your function design	
1c	Printout of your completed program code	
	Printout of the program outputs	
1d(i)	Completed task sheet explaining why output would be incorrect	
1d(ii)	Completed task sheet describing the additional refinements required	
1e	Completed task sheet evaluating the efficiency of your program	

## Task 2 – database design and development

Evidence		Tick
2a	Completed task sheet identifying two functional requirements	
2b(i)	Completed diagram showing the cardinality between entities	
2b(ii)	Completed task sheet stating the compound key	
2c	Printout of the implemented SQL statement	
	Printout of the output produced	
2d	Printout of the implemented SQL statement(s)	
	Printout of the output produced	
2e	Printout of the amended SQL statement	
	Printout of the output produced	
2f	Completed task sheet explaining the additional data required	

### Task 3 – web design and development

Evidence		Tick
3a	Completed task sheet showing your multi-level navigation structure design	
3b	Printout of the edited 'bedrooms.html' file	
3c	Printout of the edited 'styles.css' file	
3d	Printout of the edited 'quote.html' file	
3e	Completed task sheet stating why website is not fit for purpose	

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

- ◆ Check you have completed all parts of tasks 1 and either task 2 or 3.
- ◆ Label any printouts and 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

## Problem description

A research organisation currently gathers and stores data on the salaries paid to the chief executive officers (CEOs) of the top 100 technology companies in the country. They want a program to help them use this data effectively.

## Purpose

A CSV file stores the following data about the 100 companies:

- ◆ company name
- ◆ number of employees
- ◆ salary paid to CEO

This data will be read into parallel arrays.

The program will allow the user to enter the name of a company to find and display the difference between that company's CEO's salary and the highest paid CEO of all 100 companies.

The program will also find the highest number of employees employed by a single company, and the number of companies who employ within 10% of that figure.

Examples of the program outputs are shown below.

```
Enter the name of the company you would like to check:
```

```
Goldman
```

```
GameGo company has the highest paid CEO.
```

```
The Goldman CEO earns £222 817 less than the highest paid CEO.
```

```
The highest number of employees employed by a single company is 888.  
11 companies employ within 10% of 888.
```

## Assumptions

- ◆ the external file is current and updated regularly

## Task 1: software design and development (part A)

- 1a Using the problem description, identify the missing functional requirements of the program.

(2 marks)

<p><b>Input(s)</b></p> <ul style="list-style-type: none"><li>◆ Read the company name, number of employees and CEO salary from the file.</li></ul>
<p><b>Process(es)</b></p> <ul style="list-style-type: none"><li>◆ Find the highest CEO salary.</li><li>◆ Search for the CEO salary for chosen company.</li><li>◆ Find the highest number of employees employed by a single company.</li><li>◆ Calculate how many companies employ within 10% of the highest number of employees.</li></ul>
<p><b>Output(s)</b></p> <ul style="list-style-type: none"><li>◆ Display the name of the chosen company, the name of the company with the highest CEO salary and the difference between the two salaries or display “Company not found”.</li><li>◆ Display the number of companies who employ within 10% of the highest number of employees.</li></ul>

- ◆ 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)

Your teacher or lecturer will provide you with a CSV file called 'companies.csv'.

The file has data for 100 companies.

Each line of the CSV file stores the company name, the number of employees and the CEO salary as shown below:

```
Fidelity,319,582235
iShares,853,523579
Aviragen,548,636367
Aviragen Therapeutics,501,630486
Southern,269,369821
Southern First,141, 791140
Southern First Bancshares,623, 304523
Goldman,440,850580
Nuveen,599,834853
...
```

### Program top level design (pseudocode)

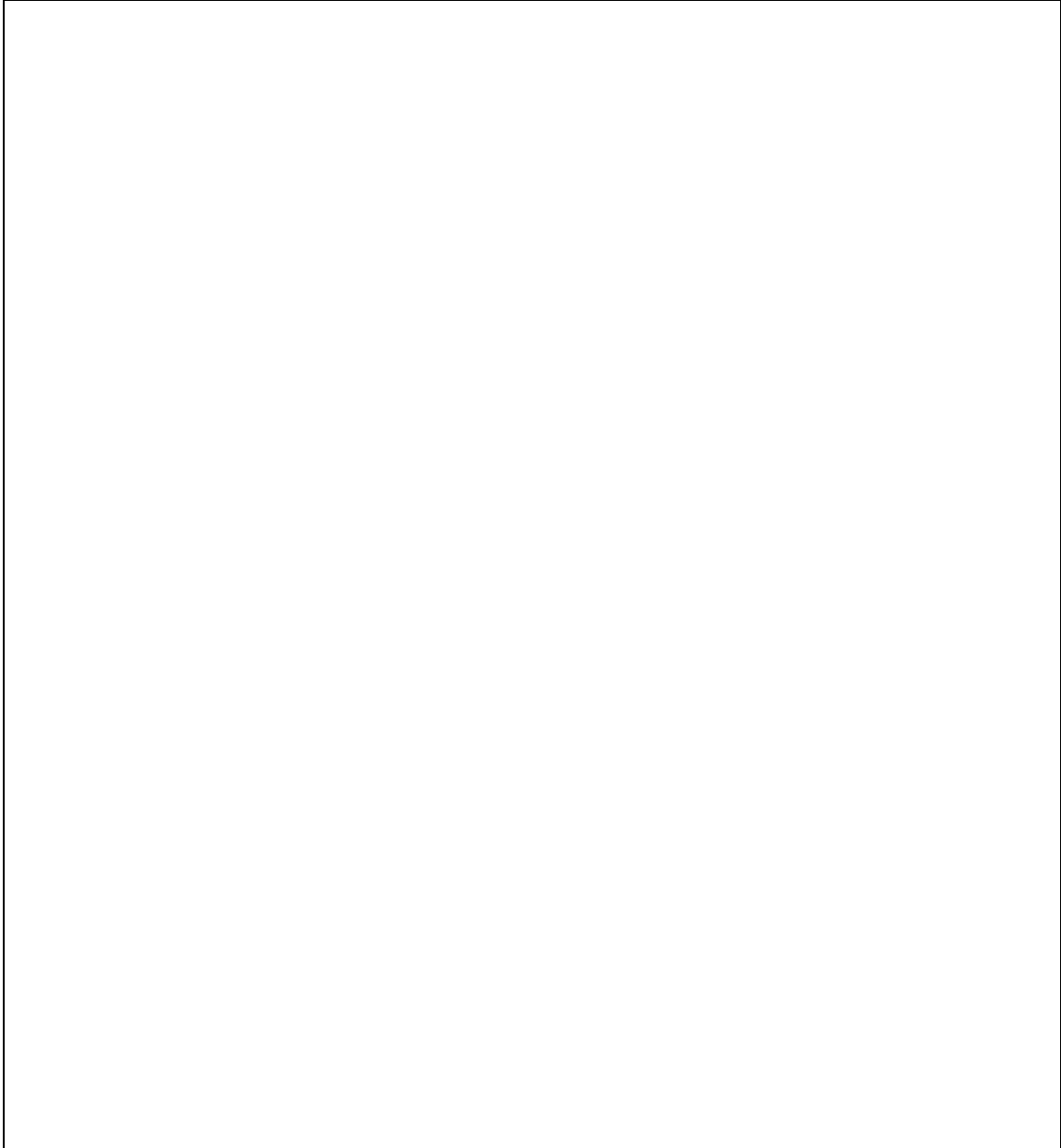
A top level design for the main steps of the program is shown below.

Steps 2 and 3 will call the same function to return the position of the highest value.

1	Read from file into parallel arrays.	OUT: company(), numEmployees(), ceoSalary()
2	Find and display the difference between the chosen company's CEO salary and the highest CEO salary.	IN: company(), ceoSalary()
3	Find and display the highest number of employees employed by a single company, and the number of companies who employ within 10% of that figure.	IN: numEmployees()

1b Using a recognised design technique, design a function called `findMaxPos` that will return the **position** of the highest value in an array. This function will be used in steps 2 and 3 of the program.

**(4 marks)**



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

Candidate name \_\_\_\_\_ Candidate number \_\_\_\_\_



# Task 1: software design and development (part C)

A top level design for the main steps of the program (with partial refinements) is shown below.

1	Read from file into parallel arrays.	OUT: company(), numEmployees(), ceoSalary()
2	Find and display the difference between the chosen company's CEO salary and the highest CEO salary.	IN: company(), ceoSalary()
3	Find and display the highest number of employees employed by a single company, and the number of companies who employ within 10% of that figure.	IN: numEmployees()

## Refinements

- 2.1 Ask user to enter the name of chosen company
- 2.2 Set found to false
- 2.3 Call findMaxPos function to return the position of highest CEO salary
- 2.4 Loop for company array
  - 2.5 If current company is the chosen company
    - 2.6 Set found to true
    - 2.7 Set position to current index
  - 2.8 End if
- 2.9 End loop
- 2.10 If chosen company name is in list
  - 2.11 Subtract and store chosen company's CEO salary from highest CEO salary
  - 2.12 Display message containing name of company with highest CEO salary, name of chosen company, and difference in salaries
- 2.13 Else
  - 2.14 Display "Company not found"
- 2.15 End if
  
- 3.1 Call findMaxPos function to return position of highest number of employees
- 3.2 Set count to 0
- 3.3 Loop for numEmployees array
  - 3.4 If current employees is greater than or equal to maximum employees\*0.9
    - 3.5 Set count to count + 1
  - 3.6 End If
- 3.7 End Loop
- 3.8 Display message showing number of companies that employ within 10% of the highest number of employees

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

You should:

- ◆ use a single function to find and return the position of the highest CEO salary, and the position of highest number of employees
- ◆ use procedures to:
  - read data from the 'companies.csv' file to parallel arrays
  - find and display the difference between the chosen company's CEO salary and the highest CEO salary
  - find and display the highest number of employees employed by a single company, and the number of companies who employ within 10% of that figure.
- ◆ test your program by using the chosen company Goldman

**(15 marks)**

Print evidence of:

- ◆ your program code
- ◆ program outputs from 1(c)

Include your name and candidate number on all evidence.

1d Step 2 of the program is tested with the following sample test data.

```
Grap,724,375000  
Ver,163,1031000  
Meto,728,816000  
TelTo,252,1031000  
Selop,555,842000  
Sever,307,569000  
Lehar,805,564000  
EastA,401,320000
```

(i) The refinement at 2.12 is shown below:

2.12 Display message containing name of company with highest CEO salary, name of chosen company, and difference in salaries.

Explain why the output from the refinements provided for step 2 would be incorrect if the sample test data was used with Selop as a chosen input.

(1 mark)

(ii) Describe the additional refinements that would be required before step 2.12 to ensure that the correct company name(s) are found.

(2 marks)

Candidate name \_\_\_\_\_ Candidate number \_\_\_\_\_

1e Evaluate the efficiency of your own program, with reference to the use of the `findMaxPos` function.

(1 mark)

Candidate name \_\_\_\_\_ Candidate number \_\_\_\_\_

## Task 2: database design and development (part A)

The National Swimming Association (NSA) collects data of multiple events, races, teams, and swimmers. The NSA needs a database to store and manage this data and has approached a database developer to help.

During a meeting with the database developer, the end users provide the following feedback:

- ◆ “I would like to see the results of a race displayed by position, from first to last.”
- ◆ “I need to see the time difference between the winning time and each swimmer’s time for a race.”
- ◆ “I need to find the team reference number for a swimmer.”
- ◆ “I need to be able to see all the swimmers who finished first, second and third in each race.”
- ◆ “I need to know the total number of days each city has hosted an event.”
- ◆ “I need to be able to update the details of events, races and swimmers.”

2a The database exists with the appropriate entities and all appropriate attributes.

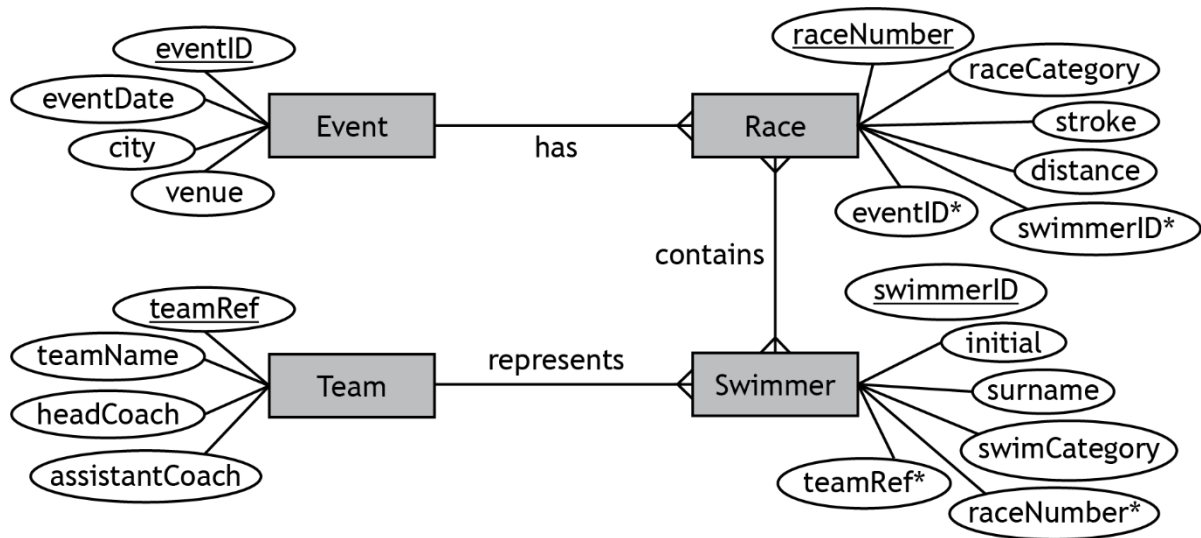
Using the user requirements above, create two functional requirements.

(2 marks)

Functional requirement 1
Functional requirement 2

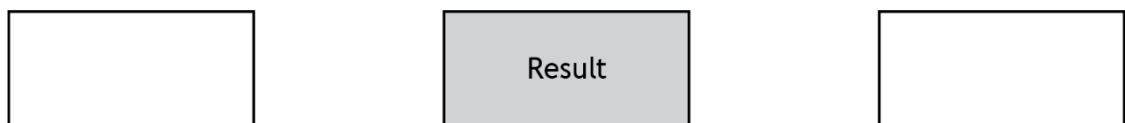
Candidate name \_\_\_\_\_ Candidate number \_\_\_\_\_

An initial entity-relationship diagram for the NSA database is shown below.



The developer advises that a fifth entity called Result must be added to the database to allow users to analyse data about what position each swimmer finished in a race.

2b (i) Complete the diagram below showing the cardinality between the Result entity and the two existing entities.



(1 mark)

(ii) State the compound key for the Result entity.

(1 mark)

- ◆ 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)

Your teacher or lecturer will provide you with the completed database.

This is a relational database with the following tables.

### NSA database

Event	Race	Result	Swimmer	Team
<u>eventID</u>	<u>raceNumber</u>	<u>raceNumber*</u>	<u>swimmerID</u>	<u>teamRef</u>
eventDate	raceCategory	position	initial	teamName
city	stroke	lane	surname	headCoach
venue	distance	<u>swimmerID*</u>	swimCategory	assistantCoach
	eventID*	raceTime	teamRef*	

2c The NSA would like to know the total number of races won by individual swimmers.

Implement the SQL statement to produce the output shown in the table below.

(4 marks)

initial	surname	swimCategory	teamName	Races won
A	Jackson	Intermediate	England	2
C	Jones	Advanced	Wales	1
D	Chaudhry	Intermediate	England	1
F	Adams	Advanced	England	4
I	Arthur	Intermediate	Scotland	2
L	Kelly	Advanced	Northern Ireland	3
M	Abbott	Intermediate	England	2
V	Rose	Advanced	Scotland	3
W	Hudson	Advanced	Wales	2

Print evidence of the implemented SQL statement and the output produced.

- 2d They want to identify the swimmer who swam in lanes 1 or 8 with the fastest time from any race.

Implement the SQL statement(s) to produce the result below.

(4 marks)

initial	surname	teamName	city	eventDate
L	Bishop	Scotland	Glasgow	06/01/2024

Print evidence of the implemented SQL statement(s) and the output produced.

- 2e All swimmers who finish in positions 1, 2 and 3 are awarded medals.

The medal total for each team is shown below.

teamName	Total medals won
England	18
Scotland	16
Northern Ireland	14
Wales	12

The following query is designed to count and display the number of medals won by each team.

```
SELECT teamName, COUNT(position) AS [Total medals won]
FROM Result, Swimmer, Team
WHERE Result.swimmerID = Swimmer.swimmerID AND Swimmer.teamRef =
Team.teamRef
GROUP BY teamName;
```

The query to test the above SQL statement is provided with the database. When run, the actual output does not match the expected output.

Amend the query to produce the expected output.

(2 marks)

Print evidence of the amended SQL statement and the output produced.



2f The end-user requirement below could not be met.

“I need to know the total number of days each city has hosted an event.”

Explain, with reference to the database structure, what additional data would be required.

**(1 mark)**



Candidate name \_\_\_\_\_ Candidate number \_\_\_\_\_

## Task 3: web design and development

HandiMaxi is a renovations company. It wants to create a website to promote its business and attract more customers.

They commissioned a web developer to produce the website. On completing the analysis stage, the developer identifies the following end-user and functional requirements.

### End-user requirements

Customers want to:

- ◆ read information about HandiMaxi
- ◆ see pictures of the work carried out by HandiMaxi
- ◆ find out information about different types of bedrooms, bathrooms and kitchens
- ◆ read customer reviews
- ◆ get an online quote for a job

### Functional requirements

- ◆ a navigation bar with links to the main pages of the website
- ◆ the 'Home' page should display:
  - information about HandiMaxi
  - a sample of positive customer reviews
- ◆ the 'Bedrooms' page should display:
  - information about Luxury, Contemporary and Modern bedrooms, displayed when the relevant button is selected
- ◆ the 'Bathrooms' page should display:
  - information about types of bathrooms
  - links to the Family Bathroom, En-Suite, and Wet Room pages
- ◆ the 'Kitchens' page should display:
  - information about the kitchen design service
  - images of kitchens before work had been carried out
  - images of kitchens after work had been carried out
- ◆ the 'Get a quote' page should contain a form to allow customers to:
  - get an online quote for a job
  - choose the room(s) they would like a quote for

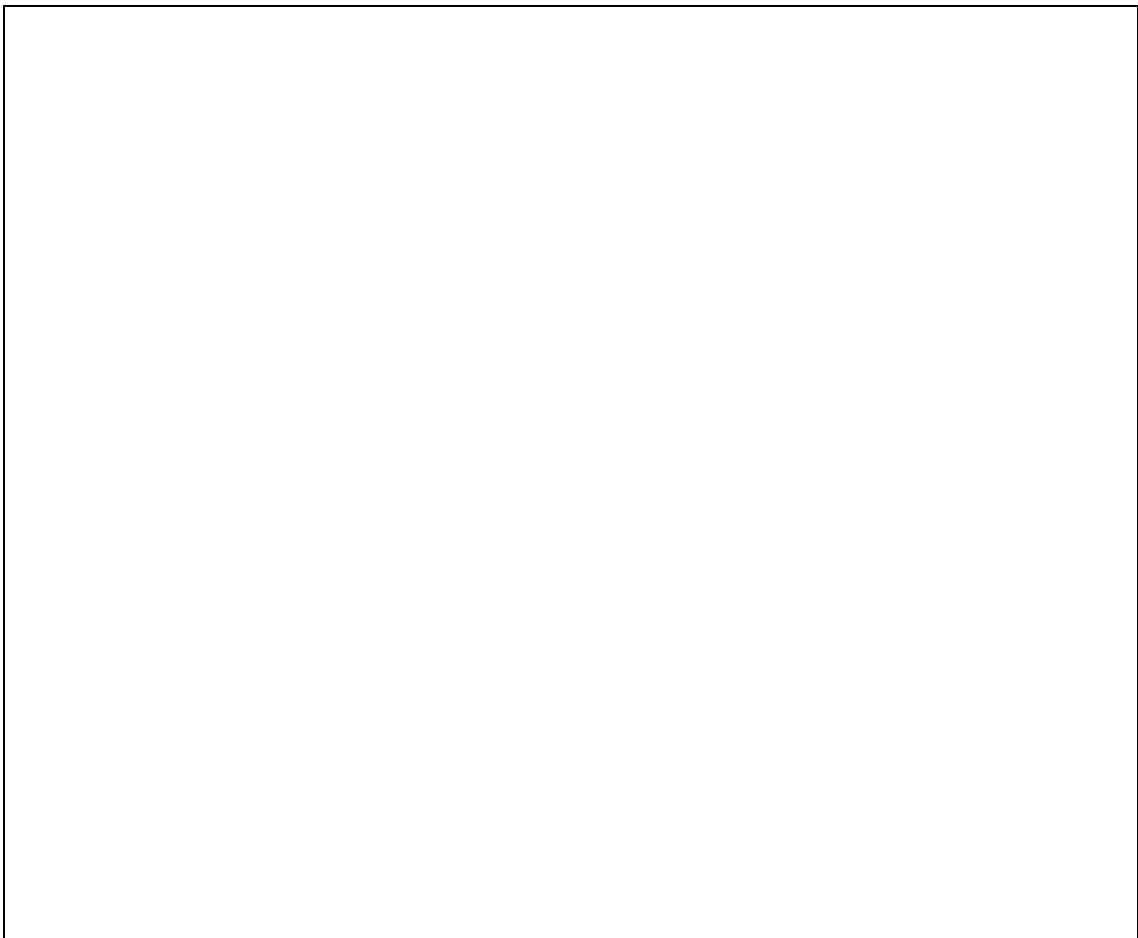
## Task 3: web design and development (part A)

- 3a The website will have a multi-level navigation structure, consisting of a 'Home' page with a horizontal navigation bar that links four main web pages: 'Bedrooms', 'Bathrooms', 'Kitchens' and 'Get a quote'.

The 'Bathrooms' page will have links to three sub-pages: 'Family Bathroom', 'En-Suite' and 'Wet Room'.

Design a multi-level navigation structure for this website, clearly showing the navigation bar and associated pages.

**(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.

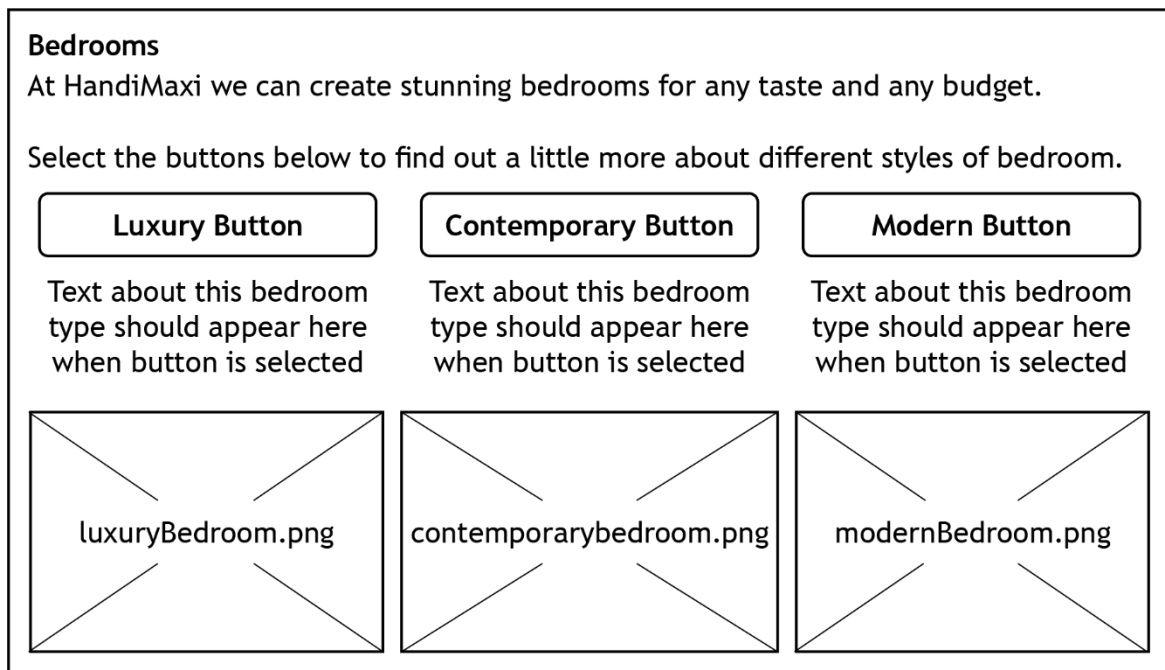
Candidate name \_\_\_\_\_ Candidate number \_\_\_\_\_

## Task 3: web design and development (part B)

Your teacher or lecturer will provide you with a copy of HandiMaxi's incomplete website.

Open the 'Home' page in a browser. Examine the home page and each of the other pages in the website.

Interactivity will be added to the 'Bedrooms' page. The description for each type of bedroom should only appear when the relevant button is selected, as shown in the wireframe below.



Open the 'bedrooms.html' file in a suitable editor.

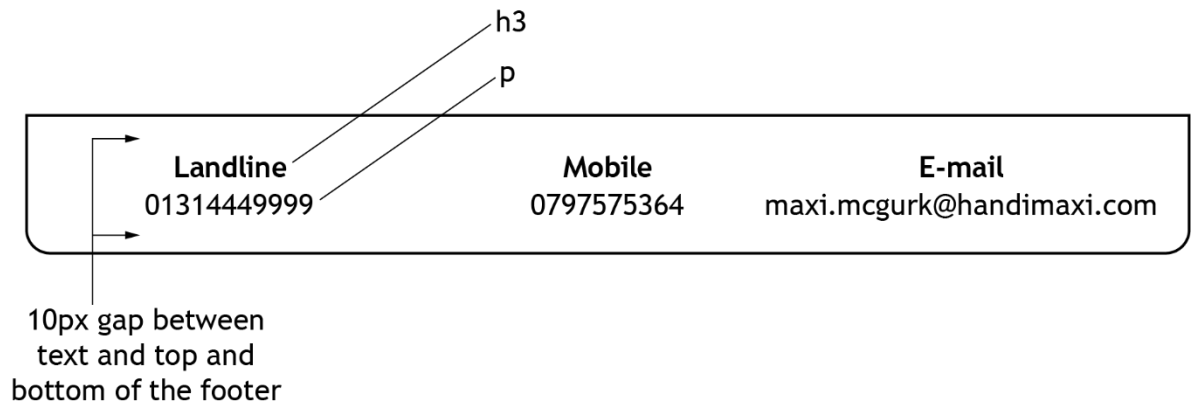
3b Edit the 'bedrooms.html' file by adding events to:

- ◆ hide all the descriptions when the bedrooms page loads
- ◆ show the description for a bedroom type when the button is selected

(4 marks)

3c Locate the 'Footer Styles' section in the 'styles.css' file.

Edit the CSS file to implement the footer as shown below. **Note:** the footer must appear on all pages of the website.



(3 marks)

Print evidence of the edited 'styles.css' file showing the CSS for navigation bar.

3d The wireframe design for the 'Get a quote' page is shown below.

Get a quote

Fill in the form below and we will get back to you ASAP

\*indicates cannot be left blank

Name:*	Description of job:*
Text input box - limited to 40 characters	Text area box - rows 19 - columns 55
Email address:*	
Text input box - limited to 40 characters	
Room(s) needing work:*	
Drop down menu to pick the room(s) (Bedroom, Bathroom, Kitchen, Somewhere else) Size 3 - Multiple	
Get a quote	

After testing, it was found that the form did not match the design.

Using the wireframe above, edit quote.html to match the design.

**(3 marks)**

Print evidence of the edited 'quote.html' file

3e State two reasons why the HandiMaxi website is not fit for purpose.

(2 marks)



# Copyright acknowledgements

Electronic files:

pics721/Shutterstock.com  
pics721/Shutterstock.com  
Artazum/Shutterstock.com  
Joseph Hendickson/Shutterstock.com  
Crystal Farmer/Shutterstock.com  
Artazum/Shutterstock.com  
Allison H Smith/Shutterstock.com  
Gabo\_Arts/Shutterstock.com  
Ground Picture/Shutterstock.com  
Gamespirit/Shutterstock.com  
Andy Dean Photography/Shutterstock.com  
Nadezhda79/Shutterstock.com  
Followtheflow/Shutterstock.com  
Gabo\_Arts/Shutterstock.com

bathroomBanner - SQA has made every effort to trace the owners of copyright of this item and seek permissions. We are happy to discuss permission requirements and incorporate any missing acknowledgement.

Please contact [question.papers@sqa.org.uk](mailto:question.papers@sqa.org.uk).



# Administrative information

---

**Published:** January 2024 (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 2024