Higher
Coursework
Assessment Task

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 2019-20 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 2020 (version 1.0)

© Scottish Qualifications Authority 2020

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 50 marks out of a total of 160 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 2019-20.

The task is valid for 2019-20 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 8 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 8-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 8 hours permitted for the assignment.

Resources

Each candidate must have access to a computer system with a high-level (textual) programming language, database application and software that can create, edit and run SQL, 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.**

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: 2019-20

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 database 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. A Microsoft Access file (carServices) 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:

- Car.csv
- Customer.csv
- Job.csv
- Garage.csv

Specific instructions for database setup
The 'carServices' database includes table names, field names, primary keys and foreign keys.

You do not need to add validation to any of the fields in the database tables.

carServices database			
Garage	**Job**	**Car**	**Customer**
garageID	jobID	regNo	customerID
garageName	garageID*	make	forename
address	dateIn	model	surname
town	dateOut	year	address
postCode	regNo*	customerID*	postCode
phoneNo	cost		phoneNo

Task 1c — 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 (testing1c.txt) to add it to the database you provide to candidates.

Task 2 — part A requires candidates to analyse and design a solution to a software problem. 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. Candidates must still have access to the problem description page during part B.

Give the following data file to candidates:

- members.txt

Task 3 — part A requires candidates to analyse and design 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 file (maxPizzeria) has been provided. This contains the CSS, HTML, and the images candidates need to complete this task. These files must not be renamed and they must remain in the folders shown below.

- css
- html
- images

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 50 marks out of a total of 160 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 three short practical tasks. You may complete the tasks in any order.

Advice on how to plan your time

You have 8 hours to complete the assignment. Marks are allocated as follows:

- Task 1 — database design and development 13 marks (26% of total)
- Task 2 — software design and development 25 marks (50% of total)
- Task 3 — web design and development 12 marks (24% of total)

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

Task 1	Evidence	
Part A		
1a(i)	Completed task 1 sheet listing attributes to be stored in Car entity	☐
1a(ii)	Completed task 1 sheet identifying functional requirements	☐
Part B		
1b(i)	SQL statement implemented listing total sales by each garage	☐
	Printout of output	☐
1b(ii)	SQL statements implemented identifying the car that spent the most number of days in a garage	☐
	Printout of output from each SQL statement	☐
1c(i)	Completed task 1 sheet explaining why output does not match	☐
1c(ii)	Completed task 1 sheet describing the change needed	☐

Task 2	Evidence	
Part A		
2a	Completed task 2 sheet identifying two boundaries	☐
2b	Completed task 2 sheet showing completed data flow design	☐
Part B		
2c	Printout of program code	☐
	Printout of output showing new member added	☐
2d	Completed task 2 sheet with description of comprehensive testing	☐
2e	Completed task 2 sheet with evaluation of program code	☐

Task 3	Evidence	
Part A		
3a	Completed task 3 sheet showing completed wireframe	☐
Part B		
3b(i)	Printout of edited 'menu.html' file showing code	☐
	Printout of 'Menu' page as viewed in a browser when first loaded	☐
	Printout of 'Menu' page as viewed in a browser when the mouse moves over a pizza image	☐
3b(ii)	Printout of edited 'order.html' file showing code	☐
	Printout of 'Order' page as viewed in a browser	☐
3c	Completed task 3 sheet with description of testing	☐
3d	Completed task 3 sheet with evaluation of fitness 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, 2 and 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: database design and development (part A)

Pit Stop is a company that repairs cars. They have five garages located in Glasgow, Edinburgh, Stirling and Kilmarnock.

Pit Stop wants to create a relational database to store details of jobs carried out by each garage and have appointed a developer to do this.

The developer asked garage staff about the features they would like in the completed database. The following is a summary of their responses.

I would like to view details of customers' cars in order of the year manufactured.

I need to be able to find customers' contact details so the garage can contact them.

I want to be able to produce a list of all cars (make or model) checked in for jobs.

I need to know how many days a car is in the garage.

I want to see all current jobs including date booked in, date booked out, car registration number and job cost.

I would like to be able to give customers details of garage addresses, postcodes and phone numbers.

I would like to make changes to customer information.

1a A database needs to be created with four entities: Garage, Job, Car and Customer.

Using the end-user responses from the garage staff:

(i) List the attributes to be stored in the Car entity.

(1 mark)

(ii) Identify two functional requirements.

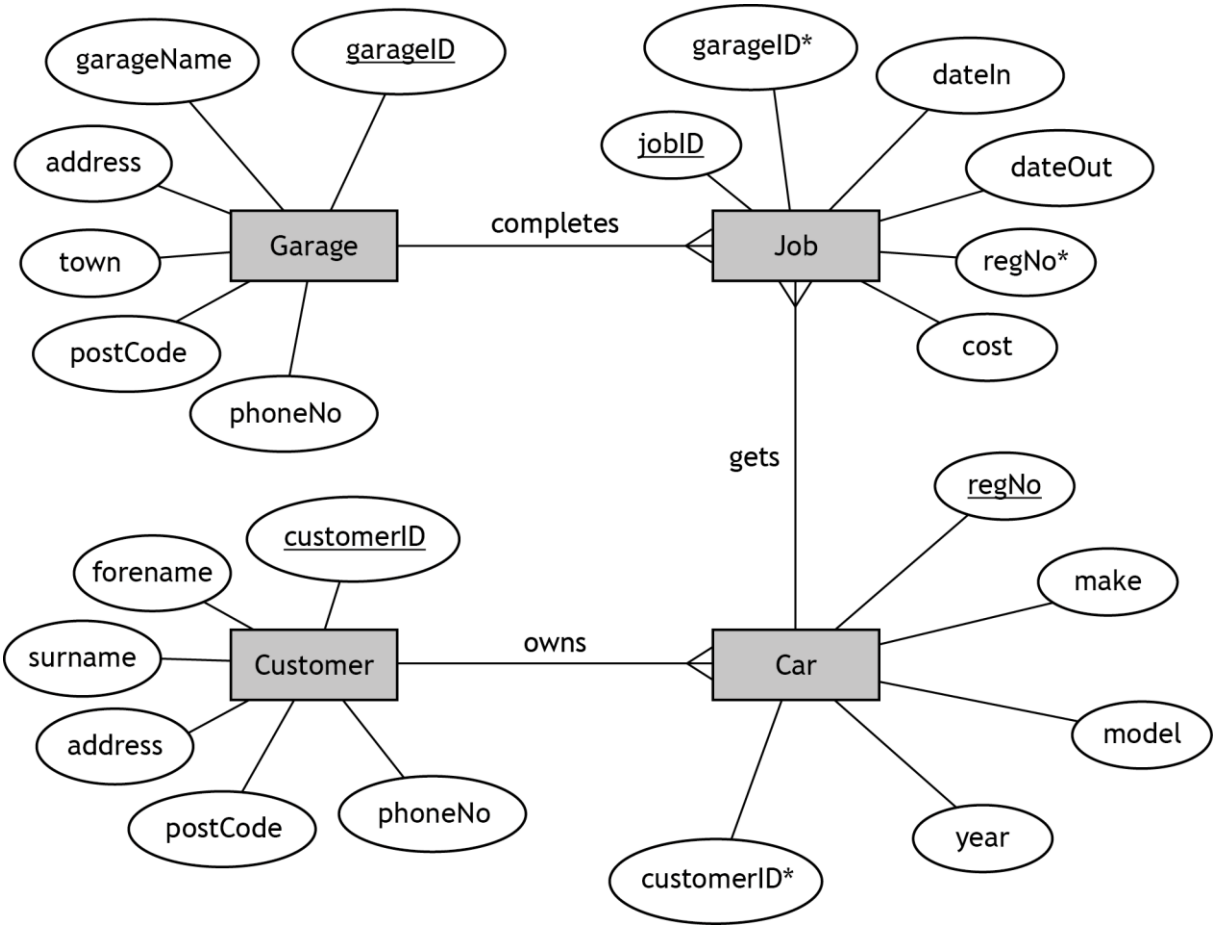
(2 marks)

- 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: database design and development (part B)

After further analysis, the developer creates the entity-relationship diagram shown below.



The design is then implemented.

Your teacher or lecturer will provide you with a completed and populated database file.

1b(i) Customers pay for completed jobs on the day they take their car out of the garage.

The company wants to list the total value of sales (in £s) for each of its five garages on
19 January 2020.

Implement the SQL statement that will produce an output with the headings:

garageName	Total sales

(4 marks)

Print evidence of:

- the implemented SQL statement
- the output it produced

Ensure your name and candidate number is on all evidence.

1b(ii) The company wants to identify the details of the car that spent the most number of days in any of its garages.

Implement **two** SQL statements that will find the highest number of days, the registration number and the name of the garage where the car was repaired.

Number of days	regNo	garageName

(4 marks)

Print evidence of:

- the implemented SQL statements
- the output produced from each statement

Ensure your name and candidate number is on all evidence.

1c The company wants to produce a list of all customers and the average cost of jobs carried out on their car(s).

They use the following SQL statement.

```
SELECT forename, surname, AVG(cost) AS [Average Job Cost]
FROM   Customer, Car, Job
WHERE  Customer.customerID = Car.customerID AND
       Car.regNo = Job.regNo
GROUP BY forename, surname
ORDER BY AVG(cost) DESC;
```

Part of the expected output is shown below.

forename	surname	Average Job Cost
Colin	Wilson	£701.10
Derek	Tsang	£657.41
Mark	Jones	£464.84
Angela	Smith	£434.49
Jennifer	Hart	£414.31
Angela	Smith	£408.85
Colin	Wilson	£249.99
Mark	Jones	£240.72
…	…	…

A query to test the above SQL statement is provided with the database.

Test the SQL statement by running the query then answer the questions on the next page.

(i) Explain why the actual output does not match the expected output.

(1 mark)

(ii) Describe the change needed to ensure that the SQL statement produces the correct output.

(1 mark)

Candidate name_____ Candidate number_____

Task 2: software design and development (part A)

Teetastic Golf Club has a maximum of 50 members. It would like to develop a program to manage the information of existing and new members.

Purpose
When a new member visits the golf club, an administrator enters the member's first name, surname and category of membership into the program. During this process, the new member enters a password that is stored along with their details. The password is then validated to ensure that it meets password strength requirements.

The program reads the existing member data from a text file and the new member's details are added to the existing member details. Details of all members are then displayed.

The program can find and display the number of members in each category (junior, adult or senior) and the total number of members.

Functional requirements
Inputs
- new member first name, surname, category and password from keyboard
- existing member first name, surname, category and password from file

Processes
- validate password
- store each member's details in appropriate data structures
- find the number of members in each category (junior, adult or senior)
- find the total number of members

Outputs
- display name, surname and category of all members
- display the number of members in each category
- display the total number of members

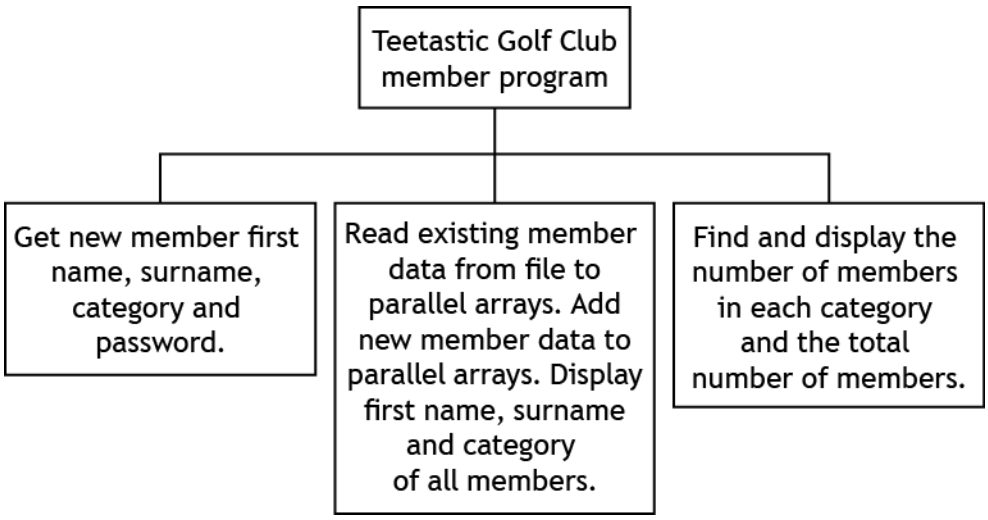
There is no requirement to write data back to file.

2a Using the information provided, identify two boundaries of this problem.

(2 marks)

Candidate name_____ Candidate number_____

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



2b Complete the table below to show data flow for the program.

(3 marks)

Top level design main program modules		
Get new member first name, surname, category and password.	IN	
	OUT	
Read existing member data from file to parallel arrays. Add new member data to parallel arrays. Display first name, surname and category of all members.	IN	
	OUT	
Find and display the number of members in each category and the total number of members.	IN	category()
	OUT	

- 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: software design and development (part B)

The design for the Teetastic Golf Club member program is shown below.

Program top level design (pseudocode)

1. Get new member first name, surname, category and password.	OUT: first name, surname, category, password
2. Read existing member data from file to parallel arrays. Add new member data to parallel arrays. Display first name, surname and category of all members.	IN: first name, surname, category, password OUT: category()
3. Find and display the number of members in each category and the total number of members.	IN: category()

Refinements

1.1 Get first name
1.2 Get surname
1.3 Get category
1.4 Call function to get a valid password OUT: password

1.4.1 Loop until password is valid
1.4.2 Ask the user to enter a password
1.4.3 Check that the first character is a capital letter (ASCII values 65 to 90)
1.4.4 Check that the last character is #, $ or % (ASCII values 35 to 37)
1.4.5 Return a valid password

2.1 Read existing member data from file into four parallel arrays: firstName(), surname(), category(), password()
2.2 Add the new member data to the existing member data in the parallel arrays
2.3 Display the first name, surname and category of all members

Your teacher or lecturer will provide you with a file “members.txt” that contains the data for 10 existing members of the club. The maximum number of members is 50.

2c Using the data flow, refinements and the information provided, implement the program in a language of your choice. Your programming language may need you to initialise variables before step 1 of the design.

Your fully completed program should:

- use a procedure to get new member data
- use a function to validate and return the new member password
- use a procedure to read existing member data from file, add new member data to parallel arrays, display member details and return the category array
- use a procedure to find and display the number of members in each category and the total number of members
- match the top level design provided
- be maintainable and modular

Check that your program is working by adding the following new member data.

Oliver Wilson, Adult, Ninjago$

The expected output from the finished program is shown below.

```
Our members are:
Angela Rich Adult
Siraj Adkins Junior
Stefano Love Senior
Cameron Wilder Junior
Griff Sutherland Adult
Amaan Sosa Senior
Isaak Schroeder Junior
Nana Galloway Junior
Lila Blanchard Adult
Eren Acosta Adult
Oliver Wilson Adult
There are currently 5 Adult members
There are currently 4 Junior members
There are currently 2 Senior members
Total current membership is  11
```

(15 marks)

Print evidence of:

- your program code
- the program output showing the new member data

Ensure your name and candidate number is on all evidence.

2d Describe how the function being used to validate the password could be comprehensively tested.

Your answer should make reference to your code.

(2 marks)

2e With reference to your own program code, evaluate:

the efficiency of your program	**(1 mark)**
the robustness of your program	**(1 mark)**
the fitness for purpose of your program	**(1 mark)**

Candidate name_____ Candidate number_____

Task 3: web design and development (part A)

Max has his own pizza shop called Max Pizzeria. He wants to create a website to promote his business and to enable customers to order their pizza online.

Max hires 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:

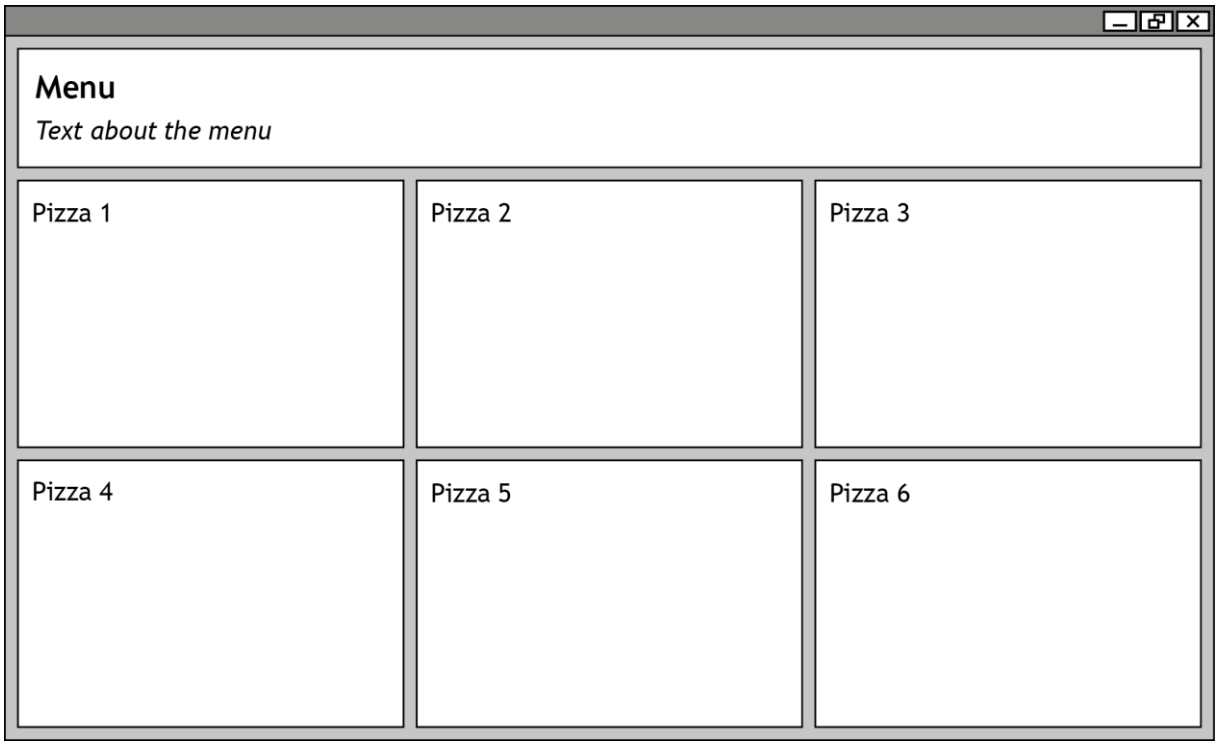
- information about Max Pizzeria
- to see pictures of the pizzas available on the menu
- to view the toppings of the pizzas on the menu
- to read customer reviews
- to search for reviews by star rating
- to add their own review to the website
- to order pizza online
- to add additional toppings to a pizza when they are ordering online
- to choose to collect or have their pizza delivered
- an option to download the Max Pizzeria app

Functional requirements

- the 'Home' page should contain:
  - information about Max Pizzeria
  - a link to download the app
- the 'Our Pizza' page should contain:
  - information about Italian pizza
  - information about where the ingredients are sourced
- the 'Menu' page should contain:
  - names and images of the pizzas available
  - a list of pizza toppings displayed to the left of each image
- the 'Reviews' page should contain:
  - customer reviews of Max Pizzeria
  - a list of reviews in descending order by star rating
  - a feature to search for reviews by star rating
  - a form to allow customers to add a new review
- the 'Order' page should contain a form to allow customers to:
  - order pizza and additional toppings online
  - choose delivery or collection

3a Complete the wireframe design below to show the position of text and images for the 'Menu' page.

(2 marks)



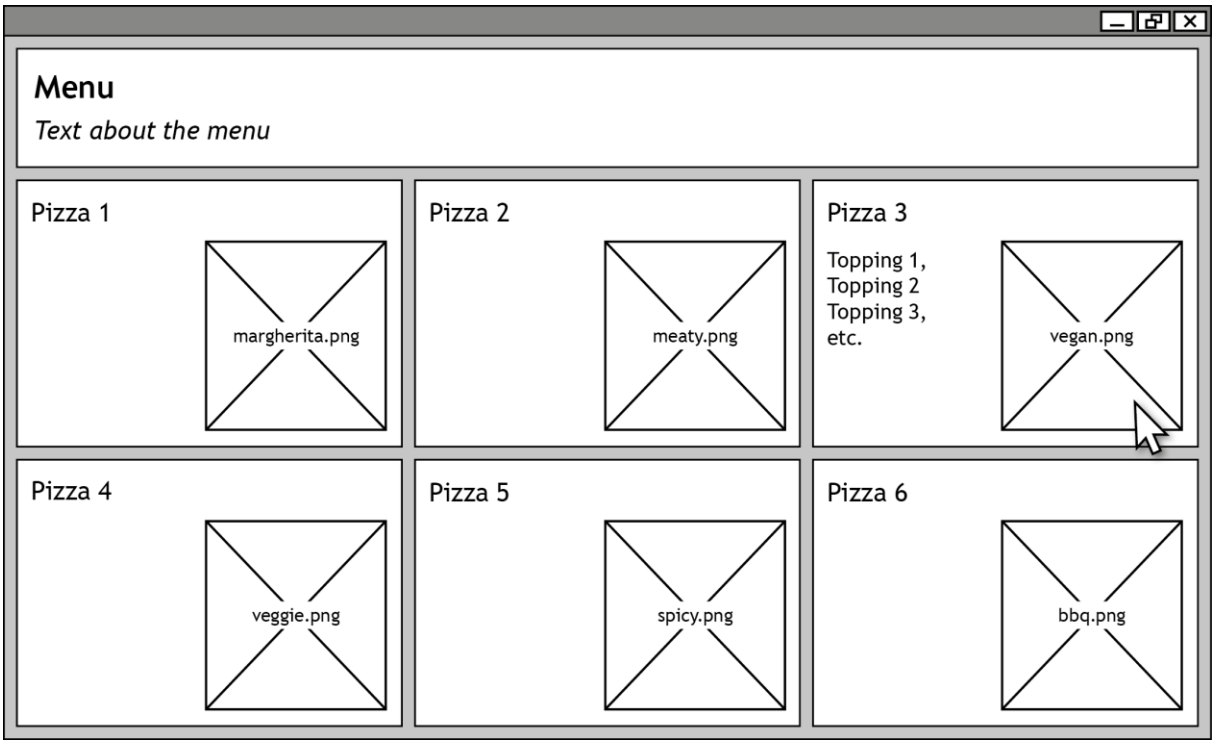
- Check your design carefully. You cannot return to part A after you hand it in.
- When you are ready, hand part A to your teacher or lecturer and collect part B.

Candidate name_____ Candidate number_____

Task 3: web design and development (part B)

After looking at the wireframe, the developer decides to make the menu page interactive. The toppings now only appear when the mouse is moved over each image.

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



3b(i) Open the 'Home' page in a browser. Examine the home page and each of the other pages in the website.

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

Add your name and candidate number to the `<footer>` element of the page.

Edit the 'menu.html' file by adding events to:

- hide all the toppings when the menu page loads
- show the toppings for a pizza when the mouse is moved over the corresponding image
- hide the toppings for a pizza when the mouse is moved away from the corresponding image

(2 marks)

Print evidence of:

- the edited 'menu.html' file
- the 'Menu' page, as viewed in a browser when first loaded
- the 'Menu' page, as viewed in a browser when the mouse moves over a pizza image

3b(ii) The website has to contain an order form for customers. The wireframe design for the 'Order' page is shown below.

Fill out the form below and get yourself a tasty pizza tonight

Your Details:

Name:* text input box - max size 30 characters

Address (including post code):*

text area box - rows 5 - columns 50

Contact No. (No Space):* text input box - max size 11 characters

Your Order:

Choose your pizza: drop down menu to pick the pizza type (Margherita, Meaty, Vegan, Veggie, Spicy, BBQ)

Extra topping(s): drop down menu to choose the topping(s) (Cheese, Chicken, Pepperoni, Ham)

size 3 - multiple

Quantity (between 1 and 5): number input box – number in range 1 to 5

Delivery or Collection:

◉ Delivery ○ Collection

Add to Basket

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

Add your name and candidate number to the `<footer>` element of the page.

Edit the file to complete the form as shown on the wireframe design.

(5 marks)

Print evidence of:

- the edited 'order.html' file
- the 'Order' page as viewed in a browser

3c Describe how the 'Extra topping(s)' field would be tested.

(1 mark)

3d State two reasons why the Max Pizzeria website is not fit for purpose.

(2 marks)

Candidate name_____ Candidate number_____

Acknowledgement of copyright
Electronic files -
Task 3 - adidas4747/Shutterstock.com
Task 3 - 4 PM Production/Shutterstock.com
Task 3 - Ratov Maxim/Shutterstock.com
Task 3 - Irina Rogova/Shutterstock.com
Task 3 - About life and fashion/Shutterstock.com
Task 3 - Dmitry Rukhlenko/Shutterstock.com
Task 3 - hitforsa/Shutterstock.com
Task 3 - Prostock-studio/Shutterstock.com
Task 3 - AlexeiLogvinovich/Shutterstock.com
All other images copyright SQA

Administrative information

Published: January 2020 (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 2020