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 2018-19 only.

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

Where this publication includes material for which SQA does not own the copyright, this material must only be reproduced on a non-commercial basis for the purposes of instruction in an educational establishment. If it is to be reproduced for any other purpose, it is the user's responsibility to obtain the necessary copyright clearance from the copyright owner. The acknowledgements page lists the owners of copyright items that are not owned by SQA.

This edition: January 2019 (version 1.0)

© Scottish Qualifications Authority 2019

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. It must be read 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 National 5 Computing Science for the academic session 2018-19.

The task is valid for 2018-19 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

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 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, screenshots, must have this information added to them.

Specific instructions for teachers and lecturers: 2018-19

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 design a database. They **must** submit their evidence for this part of the task to you before starting part B.

Task 1 — part B is a separate section. This ensures that candidates do not access part A and change their responses. You should use the data dictionaries below to create a two table, relational database with no records. Add the text sizes, presence checks and range validation for the rating field.

Entity name: Vlogger					
Attribute name	**Key**	**Type**	**Size**	**Required**	**Validation**
vloggerID	PK	number		Y	
forename		text	20	Y	
surname		text	20	Y	
username		text	6	Y	
expertise		text	15	Y	

Entity name: Video					
Attribute name	**Key**	**Type**	**Size**	**Required**	**Validation**
videoID	PK	number		Y	
vloggerID	FK	number		Y	
videoName		text	30	Y	
duration				Y	
dateCreated		date		Y	
content		text	40	Y	
rating		number		Y	Range: >=1 and <=5

Candidates **must** submit their evidence for this part of the task to you before starting part C.

Task 1 — part C is a separate section. This ensures that candidates do not access part B and change their responses. A Microsoft access file (Vlogger) is provided for candidates to use in part C. If your centre uses a different database management system, you can create the relational database for part C using the CSV files provided:

- vlogger.csv
- video.csv

If using CSV files, the relational database should include only the tables, field names, validation and relationship (with referential integrity enforced).

Task 2 — part A requires candidates to analyse a problem. They must submit their evidence for this part of the task to you before starting part B.

Task 2 — part B is a separate task. This ensures candidates do not access part A and change their responses. Candidates should still have access to the problem description (page 13) during part B.

Task 3
The following files that candidates need to complete this task have been provided. These files must not be renamed.

- clothes.png
- stock.mp4

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 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	Completed task 1 sheet showing data dictionary	☐
Part B		
1b	Printout or screenshots of database table showing correct validation has been set up for Vlogger table	☐
Part C		
1c (i)	SQL statement to show the usernames and video names of all videos with a rating greater than 3.	☐
	Printout of query results	☐
1c (ii)	SQL statement to delete unwanted “Slime” video	☐
	Printout of Video table (after the deletion)	☐
Task 2	**Evidence**	
Part A		
2a	Completed task 2 sheet showing the program analysis	☐
Part B		
2b	Printout of your program code	☐
2c	Complete task 2 sheet showing the test table	☐
2d	Printout evidence of test runs using the supplied student names	☐
2e	Completed task 2 sheet showing evaluation	☐
Task 3	**Evidence**	
3a	Completed task 3 sheet showing two functional requirements	☐
3b	Printout evidence of HTML with internal CSS file showing new page	☐
	Printout of web page as viewed in a browser	☐
3c	Completed task 3 sheet showing the description of two tests	☐
3d	Completed task 3 sheet showing evaluation	☐

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 1b, 2b)
- Clearly display your name and candidate number on each printout

Task 1: database design and development (part A)

Video bloggers (vloggers) create videos to upload to social media websites. Mirren promotes vloggers across Scotland. She keeps a record of vloggers and the details of their videos. Mirren names each video and rates them on a scale of 1 to 5 (one being the worst and five being the best). Videos may be up to 300 seconds in length.

Mirren decides to store these details in a database. The completed analysis of inputs is shown below.

Vlogger details:	Video details:
vloggerID forename surname username expertise	videoID vloggerID videoName duration dateCreated content rating

1a Complete the data dictionary for the Video entity.

(5 marks)

Entity name: Video					
Attribute name	**Key**	**Type**	**Size**	**Required**	**Validation**
videoID		number		Y	
vloggerID		number		Y	existing vloggerID from Vlogger table
videoName		text	30		
duration				Y	
dateCreated		date		Y	
content		text	40	Y	
rating		number		Y	

- 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)

1b Your teacher or lecturer will provide you with a relational database file containing two empty tables.

Using the data dictionary below complete the Vlogger table by adding the validation.
(2 marks)

Entity name: Vlogger					
Attribute name	**Key**	**Type**	**Size**	**Required**	**Validation**
vloggerID	PK	number		Y	
forename		text	20	Y	
surname		text	20	Y	
username		text	6	Y	Length=6
expertise		text	15	Y	Restricted choice: Programming, Gaming, Baking, Crafts, Makeup, Clothes

Print evidence to show that you have added both validations to the Vlogger table.

- Check your evidence 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.

Task 1: database design and development (part C)

1c Your teacher or lecturer will provide you with a completed database file including data on vloggers and videos.

(i) Mirren wants to advertise the best videos.

She wants to display the username and videoName of all videos with a rating greater than 3.

Implement the SQL statement that will output usernames and videoNames from the Vlogger and Video tables where the rating is greater than 3.

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

(4 marks)

(ii) One of the videos called "Slime" contains a recipe for slime which does not work. It should be removed from the database.

Implement the SQL statement that will delete the Slime video which has a videoID of 3.

Print evidence of your SQL statement and the Video table after the SQL statement has been implemented.

(2 marks)

Task 2: software design and development (part A)

Logan is a technician who has to generate usernames for a school's Wi-Fi service.

Logan wants to write a program that will automatically generate unique usernames for students. The usernames have to be six characters long. The program should generate and display a list of student usernames.

Program analysis

The program will ask how many usernames are to be generated. For each username, the first three letters of the student's first name will be entered and then combined with a random ending from the list below.

The program stores five endings:

ing
end
axe
gex
goh

For a student with the first name David the technician would enter Dav. The program will generate the username by joining Dav to one of the endings listed above. For example the username generated could be Daving.

2a Complete the table by filling in the missing input, process and output.

(3 Marks)

Input	
1.	
2.	Enter the first 3 letters of the student name
Process	
1.	Check length of partial student name
2.	
3.	Add the partial student name with the randomly generated ending from the stored list
Output	
1.	

- 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)

Program design

Main Steps: Pseudocode

1. Store the endings
2. Enter the number of students
3. Start fixed loop for each student
    4. Enter first three letters of student's name
    5. Generate random number
    6. Generate username
    7. Display the username
8. End Loop

REFINEMENTS

4.1 Start conditional loop

4.2 Get the first three letters of student's name

4.3 If the length of the name is not equal to 3 then

4.4 Display an error message

4.5 End If

4.6 Repeat until the name entered is 3 characters long

6.1 If the first random number was generated add the first stored ending to the end of the first three letters of the student's name

6.2 If the second random number was generated add the second stored ending to the end of the first three letters of the student's name

6.3 If the third random number was generated add the third stored ending to the end of the first three letters of the student's name

6.4 If the fourth random number was generated add the fourth stored ending to the end of the first three letters of the student's name

6.5 If the fifth random number was generated add the fifth stored ending to the end of the first three letters of the student's name

2b Using the program design and refinements, implement the program in a language of your choice. Ensure the program matches the pseudocode provided.

(15 marks)

Print evidence of your program code.

2c Your program should be tested to ensure it will only accept 3 characters.

Complete the test table below

(2 marks)

Type of test	**User input**	**Expected result**	**Actual result**
Normal		Input accepted	Printout of final output to show that input is accepted.
Exceptional		Error message displayed	Printout to show that an error message is generated.

2d Test your program using the following student names.

Chris
Christina
Christopher
Chrethe
Chrisoula
Christie

Provide evidence of the inputs and outputs to show that you have completed the test.

(1 mark)

Candidate name_____ Candidate number_____

2e With reference to your code and testing, evaluate your own program by commenting on the following:

Efficient use of programming constructs in your code. **(1 mark)**
Robustness of your completed program **(1 mark)**
The readability of your code **(1 mark)**

Evaluate the fitness for purpose of the solution

(1 mark)

Candidate name_____ Candidate number_____

Task 3: web design and development

Too Good to Throw Away is a charity clothes shop. They would like a web page to encourage donations of clothes.

It will have a:

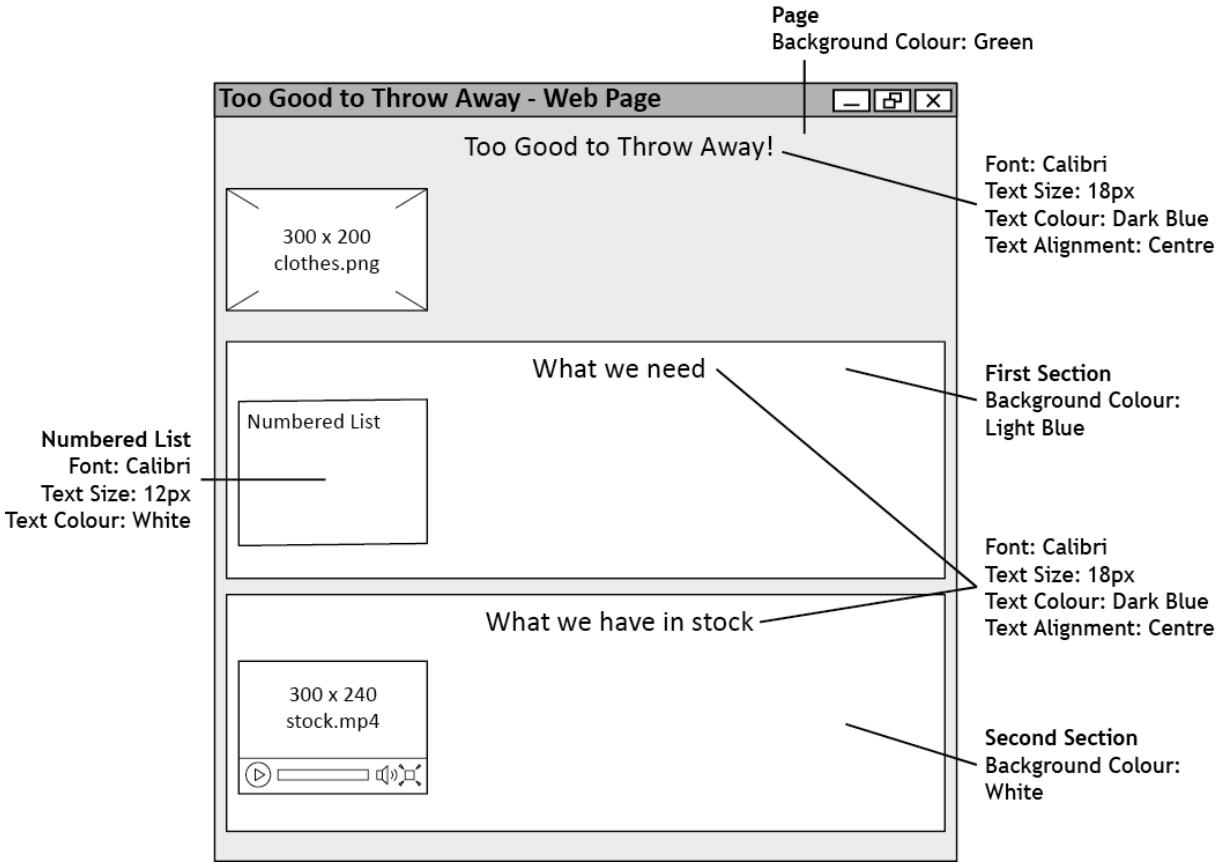
- heading with the title "Too Good to Throw Away!"
- graphic of clothes
- coloured section with a subheading entitled "What we need".
- numbered list (from 1 to 5) detailing the items the charity shop would like donated
- coloured section with a subheading titled "What we have in stock"
- video showing the current stock.

3a State **two** functional requirements for this web page.

Functional requirement 1 **(1 mark)**
Functional requirement 2 **(1 mark)**

Candidate name_____ Candidate number_____

3b. The wireframe design of the page (annotated with required styles) is shown below.



Your teacher or lecturer will provide you with the following two files:

- clothes.png
- stock.mp4

The following five items should be listed under the heading 'What we need'.

1. School Uniforms
2. Blazers
3. Gym Clothes
4. School Bags
5. Black Shoes

Implement the wireframe design using HTML and internal CSS.

(7 marks)

Print evidence of the following:

- HTML with internal CSS
- Web page as viewed in a browser

3c Describe two tests that could be performed on this web page.

Test 1 **(1 mark)**
Test 2 **(1 mark)**

3d With reference to your solution, evaluate your web page by commenting on the following:

Fitness for purpose **(1 mark)**

Candidate name_____ Candidate number_____

Acknowledgement of copyright
Electronic files -
Task 3 - STUDIO GRAND WEB/Shutterstock.com
Task 3 - Crocot/Shutterstock.com

Administrative information

Published: January 2019 (version 1.0)

History of changes

Version	Description of change	Date

Security and confidentiality

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