

The logo consists of the letters 'N5' in a large, bold, white font, set against a solid purple square background.

National 5  
Coursework  
Assessment Task



# National 5 Computing Science Assignment Assessment task

Specimen – valid from session 2017-18 and until further notice.

This edition: August 2018 (version 1.1)

The information in this publication may be reproduced to support SQA qualifications. This publication must not be reproduced for commercial or trade purposes. **This material is for use by teachers and lecturers.**

© Scottish Qualifications Authority 2012, 2018

# Contents

Introduction	1
Instructions for teachers and lecturers	2
Instructions for candidates	5
Marking instructions	18

# Introduction

This document contains instructions for teachers and lecturers, instructions for candidates, and marking instructions for the National 5 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 is a specimen assessment task.

SQA publishes a new assessment task on the secure website each academic year. The task is valid for that year 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 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 required files
- ◆ technical support

### **Evidence**

All candidate evidence (whether created manually or electronically) must be submitted to SQA in 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 Word 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 as 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, screenshots, must have this information added to them.

## **Specific instructions for teachers and lecturers: specimen assignment**

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 and design a database. They must submit their evidence for this part of the task 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 CSV file (Employee.csv) is provided for you to create a partially completed database within the application. Candidates use this to complete the task.

**Task 3** – the following files are provided and should be given to candidates to complete this task:

- ◆ winning pupil photo.jpg
- ◆ pupil interview.mp3
- ◆ pupil interview.wav

# 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 | 15 marks | (30% of total) |
| ◆ Task 2 – software design and development | 25 marks | (50% of total) |
| ◆ Task 3 – web design and development      | 10 marks | (20% 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 Word 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 are 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 (such as 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

Part A		
Task	Evidence	
1a	Completed task 1 sheet showing analysis of database inputs	<input type="checkbox"/>
1b	Completed task 1 sheet showing data dictionary	<input type="checkbox"/>
	Printout of new database table – showing new fields and their data types have been created	<input type="checkbox"/>
Part B		
Task	Evidence	
1c	Printout of screenshots – showing correct validation has been set up for the new fields	<input type="checkbox"/>
	Printout – showing evidence that a relationship exists between the two tables	<input type="checkbox"/>
1d	SQL statement to add new employee	<input type="checkbox"/>
	Printout of Employee table – showing new record	<input type="checkbox"/>
2a	Printout of your program code	<input type="checkbox"/>
2b	Completed task 2 sheet showing the two test tables	<input type="checkbox"/>
	Printout evidence of test runs – showing inputs and outputs for each of the two test tables	<input type="checkbox"/>
2c	Completed task 2 sheet showing the required test data values	<input type="checkbox"/>
2d	Completed task 2 sheet showing evaluation	<input type="checkbox"/>
3a	Completed task 3 sheet showing the end-user and functional requirements	<input type="checkbox"/>
3b	Printout evidence of HTML and CSS files – showing new page	<input type="checkbox"/>
	Printout of web page – showing how it is viewed in browser	<input type="checkbox"/>

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 1c, 2a)
- ◆ Clearly display your name and candidate number on each printout

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

Whitestar Amps design and build amplifiers for electric guitars. Below is a description of the information currently recorded for each employee and the amplifiers the employees build.

When a new employee starts working at Whitestar Amps, their first name, surname, address, contact telephone number and if they have a driving licence is recorded.

Employees build three different models of amplifier (Jazz8, Rock100 and Blues55). After each amplifier has been built and tested it is given a unique serial number which the employee enters onto a paper form. They also include the date, time of day completed, if the amplifier passed testing and their own unique employee number assigned when they were first employed.

1a Complete the missing information from the analysis of inputs below.

(2 marks)

Employee details:	Amplifier details:
	serial number date built time completed model passed test (True/False) employee number

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

1b Complete the data dictionary for the Amplifier entity.

(5 marks)

Entity name: Amplifier					
Attribute name	Key	Type	Size	Required	Validation
serialNumber		text	10	Y	length = 10
dateBuilt		date		Y	
timeCompleted				Y	
model		text	7	Y	
testPassed				Y	
employeeNumber		number		Y	existing employeeNumber from Employee table

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

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

## Task 1: database design and development (part B)

1c Using the data dictionary below complete the relational database by:

- ◆ creating a new table to store the amplifier data
- ◆ adding all required validation to fields
- ◆ creating a relationship between the two tables

(6 marks)

Your teacher or lecturer will provide you with a partially completed database file. Print evidence to show that you have completed each of the bullet points.

Entity name: Employee					
Attribute name	Key	Type	Size	Required	Validation
employeeNumber	PK	number		Y	range >=1000 AND <= 9999
firstName		text	15	Y	
surname		text	15	Y	
address		text	50	Y	
contactNumber		text	11	Y	length = 11
drivingLicence		Boolean		Y	
Entity: Amplifier					
Attribute name	Key	Type	Size	Required	Validation
serialNumber	PK	text	10	Y	length = 10
dateBuilt		date		Y	
timeCompleted		time		Y	
model		text	7	Y	restricted choice: Jazz8, Rock100 and Blues55
testPassed		Boolean		Y	
employeeNumber	FK	number		Y	existing employeeNumber from Employee table

1d The personal details of a new employee are listed below.

Employee number:	1599
Name:	Jeremy May
Address:	67 Red Lane
Driving licence:	True
Contact telephone number:	07923782534

Implement the SQL statement that will add this new record to the correct table.

**(2 marks)**

Print evidence of both the implemented SQL statement and the Employee table (clearly showing the new record).

## Task 2: software design and development

In the board game 'Capturing Olympus', six players work as a team to earn points. One point is earned if the six players score a combined total of more than 50 hits. An additional point is earned if the average number of hits is greater than or equal to 10.

Read the following analysis and design carefully.

### Program analysis

A program is required to determine the number of points earned by the team. The program will ask the user to enter the number of hits scored by each of the six players and store these values. When all six players' hits have been entered, the program will calculate the total and average number of hits. A message indicating the points earned is then displayed to the user.

### Inputs

- ◆ a valid number of hits scored by each of the six players

### Processes

- ◆ calculate the total hits achieved by all six players
- ◆ calculate an average number of hits (total/6)
- ◆ determine if the six players have earned points

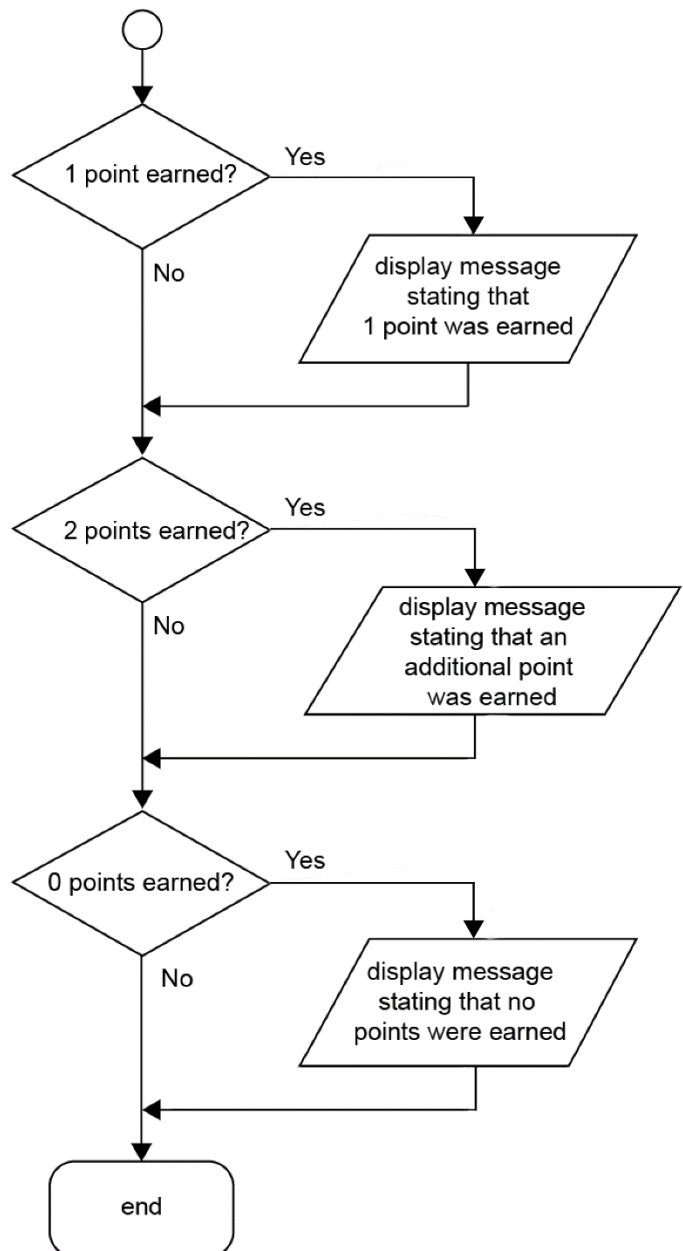
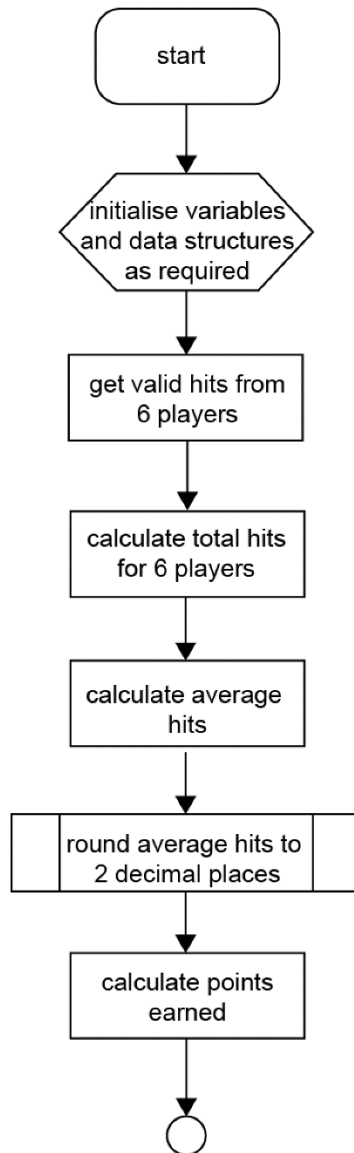
### Outputs

- ◆ a message is displayed if one point has been earned
- ◆ a message is displayed if the additional point has been earned
- ◆ a message is displayed if no points have been earned

### Assumptions

- ◆ the number of hits a single player can achieve is greater than or equal to 0 and less than or equal to 30
- ◆ the average should be displayed to two decimal places
- ◆ one point is earned if the total number of hits is greater than 50. An additional point is earned if the average number of hits is greater than or equal to 10

## Program design (flow chart)



## Task 2: software design and development

- 2a Using the program analysis and flowchart design, implement the program in a language of your choice. Ensure the program matches the design.

(15 marks)

Print evidence of the program code.

- 2b Complete the table below to create two sets of test data. You must demonstrate that the program correctly outputs the messages that one or both points have been earned.

(3 marks)

Type of test	Input	Expected output	Actual output
Normal	Player 1	Program displays message stating one point earned.	Attach printouts of inputs and outputs as evidence.
	Player 2		
	Player 3		
	Player 4		
	Player 5		
	Player 6		
Normal	Player 1	Program displays message stating two points earned.	Attach printouts of inputs and outputs as evidence.
	Player 2		
	Player 3		
	Player 4		
	Player 5		
	Player 6		

Test your program using both sets of test data. Print evidence of inputs and outputs to show that you have completed each test.

- 2c The program should ensure that only a valid number of hits can be entered for each of the six players.

State **two** extreme and **one** exceptional numerical value that could be used as part of a test run to check that only a valid number of hits can be entered:

(2 marks)

Extreme 1 \_\_\_\_\_ Extreme 2 \_\_\_\_\_

Exceptional \_\_\_\_\_

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



2d Evaluate your program by commenting on the following:

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

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

### Task 3: web design and development

Woodline Academy holds a ‘pupil of the month’ competition. They wish to add a new page to their school website each month with the following content:

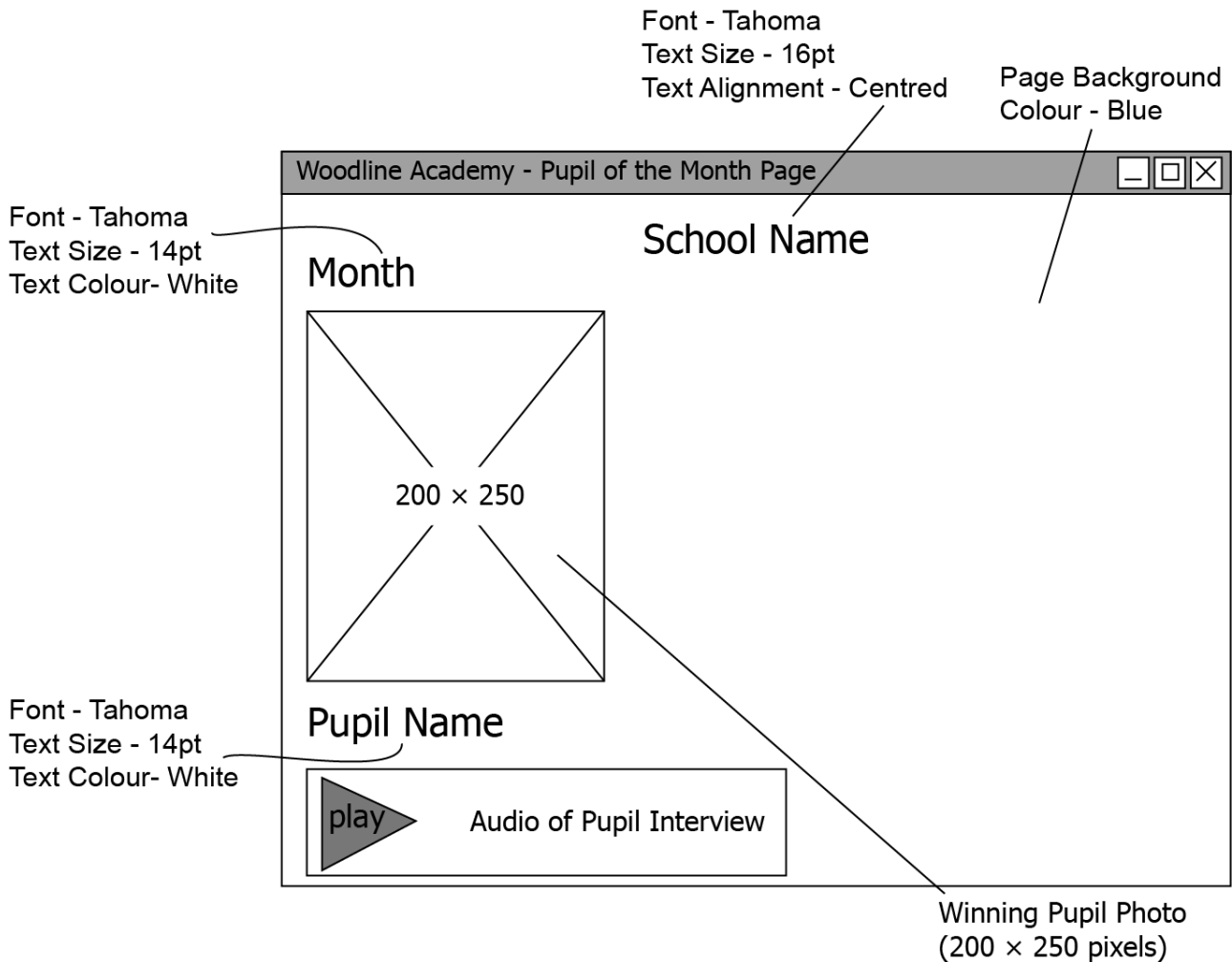
- ◆ the school name
- ◆ the month of the competition
- ◆ the name of the winning pupil
- ◆ a photo of the winning pupil
- ◆ a sound recording of an interview with the winning pupil

3a State **one** end-user requirement and **two** functional requirements for the new page.

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

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

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



You have been given the following two files:

- ◆ winning pupil photo
- ◆ pupil interview

Implement the above design using HTML and an external CSS.

**(7 marks)**

Print evidence of the following:

- ◆ HTML file
- ◆ CSS file
- ◆ Web page

# Marking instructions

Marking instructions are provided for this specimen assessment task. In line with SQA's normal practice, they are addressed to the marker. They will also be helpful for those preparing candidates for course assessment.

Marking instructions will not be provided with annual assessment tasks, as candidate evidence will be submitted to SQA for external marking.

## General marking principles

This information is provided to help you understand the general principles that must be applied when marking candidate responses in this assignment. These principles must be read in conjunction with the specific marking instructions, which identify the key features required in candidate responses.

- a Marks for each candidate response must **always** be assigned in line with these general marking principles and the specific marking instructions for this assessment.
- b Marking should always be positive. This means that, for each candidate response, marks are accumulated for the demonstration of relevant skills, knowledge and understanding: they are not deducted from a maximum on the basis of errors or omissions.

## Specific marking instructions

Task	Expected response	Additional guidance	Marks available	
<b>1</b>	<b>Database design and development – part A</b>			
1a	<p>1 mark for identifying:</p> <ul style="list-style-type: none"> <li>◆ employee number</li> </ul> <p>1 mark for identifying:</p> <ul style="list-style-type: none"> <li>◆ first name</li> <li>◆ surname</li> <li>◆ address (or street, town)</li> <li>◆ telephone number</li> <li>◆ driving licence</li> </ul>	<p>Must identify all five attributes</p>	2	Analysis (2)
1b	<p>1 mark for completion of each row:</p> <ul style="list-style-type: none"> <li>◆ serialNumber – PK</li> <li>◆ timeCompleted – time</li> <li>◆ model – restricted choice: Jazz8, Rock100 and Blues55</li> <li>◆ testPassed – Boolean</li> <li>◆ employeeNumber – FK</li> </ul>	<p>No marks awarded for application specific field types in “Type” column (for example date/time, yes/no)</p> <p>Restricted choice must list all three choices</p>	5	Design (5)

Task	Expected response	Additional guidance	Marks available
<b>1</b>	<b>Database design and development – part B</b>		
1c	1 mark each for: <ul style="list-style-type: none"> <li>◆ all fields created with correct data types</li> <li>◆ primary key (serialNumber)</li> <li>◆ presence check on every field</li> <li>◆ restricted choice on model, with correct three options</li> <li>◆ length check = 10 characters on serialNumber</li> <li>◆ linked table enforcing referential integrity</li> </ul>	Fields required: <ul style="list-style-type: none"> <li>◆ serialNumber – text</li> <li>◆ dateBuilt – date</li> <li>◆ timeCompleted – time</li> <li>◆ model – text</li> <li>◆ testPassed – Boolean</li> <li>◆ employeeNumber – FK</li> </ul>	6
1d	1 mark each for: <ul style="list-style-type: none"> <li>◆ INSERT INTO Employee</li> <li>◆ correct data, in correct order</li> </ul>	Sample answer:  INSERT INTO Employee VALUES (1599, 'Jeremy', 'May', '67 Red Lane', '07923782534',true);	2
			Implementation (8)

Task	Expected response	Additional guidance	Marks available
<b>2</b>	<b>Software design and development</b>		
2a	Array used in program		1
	Use of the following variables: <ul style="list-style-type: none"> <li>◆ total hits</li> <li>◆ average</li> <li>◆ points</li> </ul>	Variable names may differ in code  All three variables are required for 1 mark	1
	Fixed loop repeating six times (to enter player hits)		1
	Input validation – conditional loop used		1
	Input validation – correct loop condition	hits >= 0 and hits <= 30	1
	Input validation – input of player hits	Award 1 mark if not implemented within input validation loop	1
	Input validation – error message		1
	Running total calculated correctly		1
	Round function used with average		1
	Calculation of bonus points: <ul style="list-style-type: none"> <li>◆ for one bonus point</li> <li>◆ for an additional bonus point</li> </ul>		2
	Selection (if) used to display message showing one bonus point earned	(totalHits > 50) Output must be within selection	1
	Selection (if) used to display message showing additional bonus point earned	(average >= 10) Output must be within selection	1
	Selection (if) used to display message showing zero bonus points earned	(totalHits <= 50) Output must be within selection	1
	Matches design – same sequence of events as flow chart		1

Implementation (15)

Task	Expected response	Additional guidance	Marks available	
<b>2</b>	<b>Software design and development</b>			
2b	Both test tables completed to produce the required output (one bonus point or two)	Table 1 hits should total 51-59 Table 2 hits should total >=60	1	Testing (5)
	In first table, printed evidence of successful run of test data	Both inputs and outputs should be printed	1	
	In second table, printed evidence of successful run of test data	Both inputs and outputs should be printed	1	
2c	Completion of test data for input validation of player's hits for 1 mark each: <ul style="list-style-type: none"> <li>◆ extreme: 0 and 30</li> <li>◆ exceptional: any suitable, eg 1, 31</li> </ul>	Only accept numerical answers for exceptional test data	2	
2d	Evaluation of the following for 1 mark each: <ul style="list-style-type: none"> <li>◆ whether the program is fit for purpose, including explanation of code</li> <li>◆ efficient use of coding constructs</li> <li>◆ how robust the program is, including if it copes with unexpected inputs</li> </ul> Evaluation of the following for 2 marks: <ul style="list-style-type: none"> <li>◆ readability – 1 mark for each comment on the readability of the candidate's own code</li> </ul>	Efficiency answers may refer to: <ul style="list-style-type: none"> <li>◆ two loops not required for inputs and running total</li> <li>◆ single variable only required for hits if implemented in one loop</li> <li>◆ complex selection structure could have been used in place of three "ifs"</li> <li>◆ array used instead of six variables for hits</li> </ul>	5	Evaluation (5)



Task	Expected response	Additional guidance	Marks available
<b>3</b>	<b>Web design and development</b>		
3a	<p>End-user requirements could include the following for 1 mark:</p> <ul style="list-style-type: none"> <li>◆ view the winning pupil's photograph</li> <li>◆ play the winning pupil's interview</li> </ul>		1
	<p>Functional requirements could include two of the following for 1 mark each:</p> <ul style="list-style-type: none"> <li>◆ must be able to play sound</li> <li>◆ must be able to display the photograph</li> <li>◆ must be able to display the text</li> </ul>		2
			Analysis (3)

Task	Expected response	Additional guidance	Marks available
<b>3 Web design and development</b>			
3b	<p>Using the printout of the HTML file, confirm the following for 1 mark each:</p> <ul style="list-style-type: none"> <li>◆ all text and graphics content added (within structural head, title, body, tags, p, H1, div, etc)</li> <li>◆ audio tag used</li> <li>◆ link to external CSS file added in &lt;head&gt; section</li> </ul> <p>Using the printout of the CSS file, confirm the following for 1 mark each:</p> <ul style="list-style-type: none"> <li>◆ School Name Text styled correctly</li> <li>◆ Month and Pupil's Name styled using a single CSS rule</li> <li>◆ graphic size correct (CSS or HTML)</li> <li>◆ background colour changed (CSS or HTML)</li> </ul>	<p>Text and graphics checklist:</p> <ul style="list-style-type: none"> <li>◆ School Name</li> <li>◆ Month</li> <li>◆ Pupil's Photo</li> <li>◆ Pupil's Name</li> </ul> <p>CSS for School Name Text</p> <pre>h1 { font-size: 16px; font-family: "Tahoma"; text-align: center; }</pre> <p>CSS for Month/Pupil's Name Text:</p> <pre>h2 { font-size: 14px; font-family: "Tahoma"; text-color: white; }</pre> <p>CSS for &lt;img&gt; tag Internal style:</p> <pre>style="width:200px;height:250px;"</pre> <p>External stylesheet:</p> <pre>img { width:200px;height:250px;}</pre> <p>CSS for page background internal style:</p> <pre>&lt;body style="background-color:blue;"&gt;</pre> <p>External stylesheet:</p> <pre>body {background-color:blue;}</pre>	7
Implementation (7)			

# Administrative information

---

Published: August 2018 (version 1.1)

---

## History of changes

Version	Description of change	Date
1.1	'Developing skills in computer programming' changed to 'demonstrating skills in computer programming', to reflect change to course specification.  Amendments made to 'Instructions for teachers and lecturers' and 'Instructions for candidates' sections.	August 2018

## 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.

## Copyright

This document may be reproduced in whole or in part for assessment purposes provided that no profit is derived from reproduction and that, if reproduced in part, the source is acknowledged. If it needs to be reproduced for any purpose other than assessment, it is the centre's responsibility to obtain copyright clearance.

Re-use for alternative purposes without the necessary copyright clearance may constitute copyright infringement.

© Scottish Qualifications Authority 2012, 2018