



We have updated this revision support document to provide information on the modifications that were made to course assessments (coursework and exams) at the start of academic year 2021–22.

Course modifications 2021–22: National 5 Computing Science

Due to the disruption to learning caused by the COVID-19 pandemic, we made modifications at the start of the academic year. This helped reduce the volume of assessment and provide more time for learning and teaching, while maintaining the integrity and credibility of the qualifications.

The type of revision support provided in this document is dependent on the level of modifications that were made to your course assessment at the start of the year. For example, extensive modifications made at the start of academic year 2021–22, such as identifying content that would or would not be assessed, and/or removing parts of an exam paper, limit the amount of additional revision support that can be provided without undermining the integrity and credibility of the qualification. However, for every course the modifications and revision support together provide you with a significant package of help.

The type of modification(s) and revision support for this course are summarised below:

- ◆ **Modifications made at start of session:**
Information provided on question paper and assignment
- ◆ **Revision support for exam:**
Study guide

A [summary table](#) that shows the modifications we made to each course, alongside the revision support, is available on SQA's website.

We published more detailed [modification documents](#) for schools, colleges and training providers in August 2021. If you have any questions about the modifications for this course please speak to your teacher, lecturer or training provider.

Update to original revision support

As a result of ongoing quality assurance and feedback we have updated this document.

- ◆ We have removed reference to question 2016 Q20(b) from Software design and development: Implementation (computational constructs), as this was listed here in error.



2022 revision support for learners: National 5 Computing Science

Guidance for the 2022 exam

You can use the following information to help you plan your revision and prepare for the 2022 National 5 Computing Science exam.

To give you an idea of what to expect in the exam, we've provided details for some of the questions below. There will be other questions in the exam, so you need to revise all of the content for the sections you will complete.

Sections

There are three sections in the exam:

- ◆ Section 1 — Software design and development, and Computer systems (55 marks)
- ◆ Section 2 — Database design and development (25 marks)
- ◆ Section 3 — Web design and development (25 marks)

You should attempt all questions in section 1 and attempt **either** section 2 **or** section 3.

Section 1 — Software design and development, and Computer systems

For the larger questions (3–5 marks), you should be prepared to:

- ◆ design part of a program that will make decisions and perform a calculation
- ◆ use a pseudocode design to write code that uses an array and random numbers
- ◆ design the user interface for a program
- ◆ design the validation for the inputs identified in a problem
- ◆ re-write code to make it more efficient

You should also be prepared to answer smaller questions (1–2 marks) on all software design and development, and computer systems content. Some computer systems questions will be in the context of a wider software design and development question.

Section 2 — Database design and development

You should be prepared to:

- ◆ complete an entity-relationship diagram
- ◆ complete an SQL query design using multiple search criteria
- ◆ complete an SQL statement using WHERE conditions and ORDER BY

You should also be prepared to answer questions on all other database design and development content.

Section 3 — Web design and development

You should be prepared to:

- ◆ answer questions on wireframe and low-fidelity prototype web page design
- ◆ draw how a web page will look in a browser from HTML and CSS code
- ◆ use <div>, lists and classes in the implementation of HTML and CSS code

You should also be prepared to answer questions on all other web design and development content.

Exam technique

It is important you:

- ◆ read questions fully before attempting to answer them
- ◆ answer according to the question type (for example, 'identify', 'describe', 'explain')
- ◆ consider the mark allocation for each question and tailor your answer accordingly
- ◆ attempt all questions — even if you can't reach the full solution, you can still gain marks for a partially correct answer
- ◆ use the correct vocabulary and technical language
- ◆ manage the 1.5 hours available to you. The level of difficulty increases through each section. If you find yourself spending too much time on section 1, you might want to move on and complete either section 2 or section 3 (worth 25 marks and starting with some less-difficult questions), before returning to section 1.

Question types

- ◆ When answering an '**identify**', '**name**', '**give**' or '**state**' question, only a short answer is expected, this will be a single word or phrase.
- ◆ When answering a '**describe**' or '**explain**' question, you must:
 - give a fuller answer than would be expected for a 'name' or 'state' question
 - refer to the context given in the question instead of providing a generic response
 - use the number of marks available as a guide to how many points you should make in your answer

- ◆ When answering a '**write code**' question, your solution should be in recognisable code. Marks are awarded based on how you solve the problem, not the accuracy of the syntax of your code.
- ◆ When answering a '**design**' question:
 - in software design and development, you can use any recognised design technique. Marks are awarded for how you solve the problem, not for how you have used the technique. For example, you will not be penalised for the incorrect shape of box in a flowchart or structure diagram.
 - in database design and development, and web design and development, you will be asked to draw and complete designs using a specified technique. Marks are awarded for your understanding of the concepts, not for how you show these. For example, a primary key can be identified with underlining or by annotating as 'PK' or 'Primary Key' and there are many ways to label a wireframe design.

Practice questions

The best preparation for the exam is to work through questions that are similar in style.

Use the tables on the following pages to help you to identify questions for the topics you want to focus on. You can then find the questions in these documents on SQA's website:

- ◆ [specimen question paper \(SQP\) and marking instructions](#)
- ◆ [2019 past paper](#) and [marking instructions](#)
- ◆ [2018 past paper](#) and [marking instructions](#)
- ◆ [2017 past paper](#) and [marking instructions](#)
- ◆ [2016 past paper](#) and [marking instructions](#)

Note: concentrate on the questions you've identified, rather than working through the whole paper, as parts of these papers do not reflect the content and options in the 2022 exam.

Software design and development		Document	Question
Development methodologies	Describe and implement the phases of an iterative development process: analysis, design, implementation, testing, documentation, and evaluation, within general programming problem-solving.	SQP	Q2
Analysis	Identify the purpose and functional requirements of a problem that relates to the design and implementation at this level, in terms of: <ul style="list-style-type: none"> ◆ inputs ◆ processes ◆ outputs 	SQP	Q16(a)
		2019	Q13(a)
		2018	Q19(a)
Design	Identify the data types and structures required for a problem that relates to the implementation at this level, as listed below: <p>Describe, identify, and be able to read and understand:</p> <ul style="list-style-type: none"> ◆ structure diagrams ◆ flowcharts ◆ pseudocode 	SQP	Q19(a)
		SQP	Q19(b)
		SQP	Q21(c)(ii)
		2019	Q5
		2019	Q16(a)
		2019	Q16(b)
		2017	Q13
	Exemplify and implement one of the above design techniques to design efficient solutions to a problem.	2016	Q21(a)
		SQP	Q16(b)
		2019	Q16(c)
		2019	Q19(b)(ii)
		2018	Q19(d)
		2018	Q21(a)
	Describe, exemplify, and implement user-interface design, in terms of input and output, using a wireframe.	2016	Q12
2016		Q18(d)(ii)	
2019		Q3	
		2018	Q19(b)
Implementation (data types and structures)	Describe, exemplify, and implement appropriately the following data types and structures: <ul style="list-style-type: none"> ◆ character ◆ string ◆ numeric (integer and real) ◆ Boolean ◆ 1D arrays 	SQP	Q14(b)
		SQP	Q21(a)
		2019	Q13(c)
		2019	Q19(c)(ii)
		2018	Q17(a)(i)
		2017	Q3
		2017	Q15(a)
		2016	Q16(b)
		2016	Q19(a)

Software design and development		Document	Question
Implementation (computational constructs)	<p>Describe, exemplify, and implement the appropriate constructs in a high-level (textual) language:</p> <ul style="list-style-type: none"> ◆ expressions to assign values ◆ expressions to return values using arithmetic operations (addition, subtraction, multiplication, division, and exponentiation) ◆ expressions to concatenate strings ◆ selection constructs using simple conditional statements with <, >, ≤, ≥, =, ≠ operators ◆ selection constructs using complex conditional statements ◆ logical operators (AND, OR, NOT) ◆ iteration and repetition using fixed and conditional loops ◆ predefined functions (with parameters): <ul style="list-style-type: none"> — random — round — length <p>Read and explain code that makes use of the above constructs.</p>	SQP	Q4
		SQP	Q14(a)
		SQP	Q14(c)
		SQP	Q19(c)
		2019	Q13(f)
		2019	Q19(a)
		2019	Q19(c)(iii)
		2018	Q4(a)
		2018	Q7
		2018	Q11
		2018	Q14(c)
		2018	Q15
		2018	Q19(c)(i)
		2018	Q19(c)(ii)
		2018	Q21(b)(i)
		2017	Q7
		2017	Q15(c)(i)
		2017	Q19(b)
		2016	Q9
		2016	Q12
2016	Q16(d)		
2016	Q16(e)		
2016	Q18(d)(ii)		
2016	Q19(b)(i)		
2016	Q19(b)(iii)		
2016	Q19(c)		
Implementation (algorithm specification)	<p>Describe, exemplify, and implement standard algorithms:</p> <ul style="list-style-type: none"> ◆ input validation ◆ running total within loop ◆ traversing a 1D array 	SQP	Q19(e)
		2019	Q7(b)
		2019	Q13(e)
		2019	Q19(c)(i)
		2018	Q17(a)(ii)
		2018	Q21(b)(ii)
		2017	Q17(a)
		2017	Q19(a)
2016	Q16(a)		
Testing	<p>Describe, identify, exemplify, and implement normal, extreme, and exceptional test data for a specific problem, using a test table.</p>	SQP	Q21(b)
		2019	Q7(a)
		2018	Q19(e)
		2017	Q12
		2017	Q17(b)
		2016	Q16(c)
2016	Q21(c)(i)		

Software design and development		Document	Question
	Describe and identify syntax, execution, and logic errors.	SQP	Q21(c)(i)
		2019	Q11
		2019	Q19(b)(i)
		2018	Q14(a)
		2018	Q14(b)
		2017	Q17(c)
		2017	Q17(d)(i)
		2016	Q21(c)(ii)
Evaluation	Describe, identify, and exemplify the evaluation of a solution in terms of: <ul style="list-style-type: none"> ◆ fitness for purpose ◆ efficient use of coding constructs ◆ robustness ◆ readability: <ul style="list-style-type: none"> — internal commentary — meaningful identifiers — indentation — white space 	2017	Q19(c)
		2016	Q2
		2016	Q21(d)

Computer systems		Document	Question
Data representation	Describe and exemplify the use of binary to represent positive integers.		
	Describe floating point representation of positive real numbers using the terms mantissa and exponent.	SQP	Q12
		2019	Q13(d)
		2018	Q4(b)
		2017	Q2
	Convert from binary to denary and vice-versa.	SQP	Q1
		2019	Q1
		2018	Q22(a)(i)
		2017	Q15(d)
		2016	Q1
	Describe extended ASCII code (8-bit) used to represent characters.	SQP	Q6
		2019	Q13(b)(ii)
		2018	Q22(a)(ii)
		2017	Q15(c)(ii)
	Describe the vector graphics method of graphic representation for common objects: <ul style="list-style-type: none"> ◆ rectangle ◆ ellipse ◆ line ◆ polygon with attributes: <ul style="list-style-type: none"> ◆ co-ordinates ◆ fill colour ◆ line colour 	SQP	Q8
		2019	Q12
2018		Q22(b)	
2017		Q4	
Describe the bit-mapped method of graphics representation.	SQP	Q18(e)(i)	
	2019	Q13(b)(i)	
Computer structure	Describe the purpose of the basic computer architecture components and how they are linked together: <ul style="list-style-type: none"> ◆ processor (registers, ALU, control unit) ◆ memory locations with unique addresses ◆ buses (data and address) 	SQP	Q21(d)
		2019	Q10(b)
		2019	Q16(d)(i)
		2019	Q16(d)(ii)
		2018	Q21(b)(iii)
		2018	Q21(b)(iv)
		2017	Q15(b)
		2016	Q5
		2016	Q19(b)(ii)
		Explain the need for interpreters and compilers to translate high-level program code to binary (machine code instructions).	SQP
2019	Q16(e)		
2018	Q17(b)		
2017	Q15(e)		
2016	Q11		

Computer systems		Document	Question
Environmental impact	Describe the energy use of computer systems, the implications on the environment and how these could be reduced through: <ul style="list-style-type: none"> ◆ settings on monitors ◆ power-down settings ◆ leaving computers on stand-by 	SQP	Q11
		2018	Q22(c)
Security precautions	Describe the role of firewalls.	SQP	Q5(a)
		2019	Q15(d)
		2016	Q20(a)
	Describe the use made of encryption in electronic communications.	SQP	Q5(b)
		2019	Q15(d)
		2018	Q6
	2016	Q20(b)	

Database design and development		Document	Question	
Analysis	Identify the end user and functional requirements of a database problem that relates to the implementation at this level.	2019	Q14(a)	
Design	Describe and identify the implications for individuals and businesses of the UK General Data Protection Regulation (UK GDPR) that data must be:	SQP	Q3	
		2019	Q14(d)	
		2017	Q14(b)	
		2016	Q18(e)	
	<ul style="list-style-type: none"> ◆ processed lawfully, fairly and in transparent manner in relation to individuals ◆ used for declared purpose only ◆ limited to data need for the declared purpose ◆ accurate ◆ not kept for longer than necessary held securely 			
		Describe and exemplify entity-relationship diagrams with two entities indicating:	SQP	Q17(a)
			SQP	Q20(a)(iii)
			2019	Q10(a)
			2019	Q14(b)
	2018		Q23(a)	
	<ul style="list-style-type: none"> ◆ entity name ◆ attributes ◆ relationship (one to many) 			
		Describe and exemplify a data dictionary:	SQP	Q9
			SQP	Q20(a)(i)(ii)
			2019	Q8(a)
			2019	Q14(c)
			2019	Q17(a)
			2018	Q5
			2018	Q20(a)
			2018	Q20(b)
			2018	Q23(b)
2017			Q8	
2017			Q14(a)	
2017			Q14(d)	
2016			Q4	
2016			Q15(a)	
2016	Q15(d)			
2016	Q15(e)(i)			
<ul style="list-style-type: none"> ◆ attribute type: <ul style="list-style-type: none"> — text — number — date — time — Boolean ◆ attribute size ◆ validation: <ul style="list-style-type: none"> — presence check — restricted choice — field length — range 				

Database design and development		Document	Question
	Exemplify a design of a solution to the query: <ul style="list-style-type: none"> ◆ multiple tables ◆ fields ◆ search criteria ◆ sort order 	SQP	Q20(c)
		2019	Q17(b)
		2018	Q20(c)
Implementation	Implement relational databases with two linked tables, to match the design with referential integrity. Describe, exemplify and implement SQL operations for pre-populated relational databases, with a maximum of two linked tables: <ul style="list-style-type: none"> ◆ select: <ul style="list-style-type: none"> — from — where: <ul style="list-style-type: none"> ○ AND, OR, <, >, = ○ order by with a maximum of two fields ◆ insert ◆ update ◆ delete ◆ equi-join between tables Read and explain code that makes use of the above SQL.	2018	Q20(e)
		SQP	Q17(b)
		SQP	Q17(c)(ii)
		SQP	Q20(b)
		2019	Q4
		2019	Q17(c)(ii)
		2019	Q17(d)(i)
		2018	Q10(a)
		2018	Q10(b)
		2018	Q16
		Testing	Describe and exemplify testing: <ul style="list-style-type: none"> ◆ SQL operations work correctly at this level
2019	Q8(b)		
2019	Q17(d)(ii)		
2018	Q20(d)(i)		
Evaluation	Evaluate solution in terms of: <ul style="list-style-type: none"> ◆ fitness for purpose ◆ accuracy of output 	2019	Q17(c)(i)
		2018	Q20(d)(ii)

Web design and development		Document	Question	
Analysis	Identify the end user and functional requirements of a website problem that relates to the design and implementation at this level.	2019	Q15(a)	
Design	Describe and exemplify the website structure with a home page, a maximum of four linked multimedia pages, and any necessary external links.	SQP	Q15(a)	
		2019	Q18(a)	
		2018	Q18(a)	
		2017	Q20(d)	
	Describe, exemplify and implement, taking into account end-user requirements, effective user-interface design (visual layout and readability) using wire-framing: <ul style="list-style-type: none"> ◆ navigational links ◆ consistency across multiple pages ◆ relative vertical positioning of the media displayed ◆ file formats of the media (text, graphics, video, and audio) 	2016	Q17(d)(i)	
		Describe and identify the implications for individuals and businesses of the Copyright, Designs and Patents Act 1988 relating to: <ul style="list-style-type: none"> ◆ web content (text, graphics, video, and audio) 	SQP	Q18(g)
			2019	Q15(b)(i)
			2018	Q13
	2017		Q18(b)	
	Compare a range of standard file formats: <ul style="list-style-type: none"> ◆ audio standard file formats WAV and MP3 in terms of compression, quality, and file size ◆ bit-mapped graphic standard file formats JPEG, GIF, and PNG in terms of compression, animation, transparency, and colour depth 	2016	Q18(d)(i)	
		Describe the factors affecting file size and quality, relating to resolution, colour depth, and sampling rate.	SQP	Q10
			SQP	Q18(e)(ii)
			2019	Q15(b)(ii)
			2019	Q18(d)(i)
2018			Q18(c)	
2018			Q1	
2018	Q3			
Describe the need for compression.	2017	Q5		
	2017	Q16(d)(iii)		
Describe, exemplify and implement prototyping (low-fidelity) from wireframe design at this level.	SQP	Q7		
	2018	Q12		

Web design and development		Document	Question
Implementation (CSS)	Describe, exemplify and implement internal and external Cascading Style Sheets (CSS): <ul style="list-style-type: none"> ◆ selectors, classes and IDs ◆ properties <ul style="list-style-type: none"> — text: <ul style="list-style-type: none"> ○ font (family, size) ○ color ○ alignment — background colour Read and explain code that makes use of the above CSS.	SQP	Q15(b)
		SQP	Q18(d)(i)
		SQP	Q18(f)
		2019	Q15(c)(iii)
		2019	Q18(b)
		2018	Q18(d)
		2018	Q18(e)
Implementation (HTML)	Describe, exemplify and implement HTML code: <ul style="list-style-type: none"> ◆ HTML ◆ head ◆ title ◆ body ◆ heading ◆ paragraph ◆ DIV ◆ link ◆ anchor ◆ IMG ◆ audio ◆ video ◆ lists — ol, ul and li Read and explain code that makes use of the above HTML.	SQP	Q18(b)
		2019	Q2
		2019	Q9
		2019	Q15(c)(ii)
		2019	Q18(d)(ii)
		2018	Q18(b)
		2018	Q23(c)(i)
		2018	Q23(c)(ii)
		2017	Q16(c)
		Describe and implement hyperlinks (internal and external), relative and absolute addressing.	SQP
	2018		Q18(f)(i)
	2018		Q18(f)(ii)
	2017		Q1
	2017		Q16(d)(i)
2016	Q17(c)		
2016	Q17(e)		

Web design and development		Document	Question
Implementation (Javascript)	Describe and identify Javascript coding related to mouse events: <ul style="list-style-type: none"> ◆ Onmouseover ◆ Onmouseout 	SQP	Q17(c)(ii)
		SQP	Q18(c)(i)
		2019	Q15(c)(i)
		2018	Q9
		2017	Q16(b)
Testing	Describe and exemplify testing: <ul style="list-style-type: none"> ◆ matches user-interface design ◆ links and navigation work correctly ◆ media (such as text, graphics, and video) display correctly ◆ consistency 	SQP	Q15(c)
		SQP	Q18(a)
		SQP	Q18(d)(ii)
		2018	Q8
		2018	Q18(g)
		2017	Q16(e)
		2017	Q20(a)
Evaluation	Evaluate solution in terms of: <ul style="list-style-type: none"> ◆ fitness for purpose 	2019	Q6
		2019	Q18(c)
		2018	Q2

Remember to look out for *Your Exams*. This guide contains essential information and rules that you need to know about SQA exams.