Qualification Accredited

Oxford Cambridge and RSA

GCSE (9–1)

Examiners' report

COMPUTER SCIENCE

J276

For first teaching in 2016

J276/02 Autumn 2020 series

Version 1

www.ocr.org.uk/computerscience

Introduction

Our examiners' reports are produced to offer constructive feedback on candidates' performance in the examinations. They provide useful guidance for future candidates.

i	Reports for the Autumn 2020 series will provide a broad commentary about candidate performance, with the aim for them to be useful future teaching tools. As an exception for this series they will not contain any questions from the question paper nor examples of candidate answers.

The reports will include a general commentary on candidates' performance, identify technical aspects examined in the questions and highlight good performance and where performance could be improved. The reports will also explain aspects which caused difficulty and why the difficulties arose, whether through a lack of knowledge, poor examination technique, or any other identifiable and explainable reason.

A full copy of the question paper and the mark scheme can be downloaded from OCR.

Paper 2 series overview

J276/02 (Computational thinking, algorithms and programming) is one of two examination components for the GCSE Computer Science. This component focuses on:

- algorithms
- programming techniques
- producing robust programs
- computational logic
- translators and facilities of languages
- data representation.

To do well on this paper, candidates need to be comfortable with writing, completing and using algorithms using pseudocode and/or flowcharts. This may involve applying their knowledge to unfamiliar contexts.

Candidates did well where it showed that they had extensive practice of producing and completing algorithms. Using a high-level language in classroom situations allows candidates to answer questions on this paper more confidently.

The mandatory Programming Project task allows candidates to spend a significant amount of time practising their programming skills.

Section 5f of the specification shows how pseudocode and Boolean logic will be presented in examinations. Candidates should be aware of these conventions. This will allow them access and understand examination questions.

Candidates' responses are accepted in any logical form; they are not required to present responses in a specific form.

Fewer candidates took this examination when compared to previous series. The standard of answers was generally good.

Candidates who did well on this paper generally did the following:	***Candidates who did less well on this paper generally did the following:***
• Decomposed Question 6(c) and attempted to tackle each requirement; even where mistakes were made, marks were given for correct elements. • Applied their answers to the context given in the question, where applicable. For example, Question 6(a) requires candidates to describe the steps a linear search would take to find a specific given in a given array. • Showed that they were able to understand programming concepts such as data types (Question 2(a)) and parameter passing with functions (Question 3(b)(ii)) when applied to given pseudocode.	• For algorithms, rewrote the question rather than attempting to specify how a problem should be solved. • Demonstrated that they perhaps were not fully aware of the use of `SWITCH/CASE` statements as an alternative method of selection (Question 3(f)) or SQL (Question 3(e)) • Gave generic answers where a context or scenario was given in the question.

Comments on responses by question type

Questions where candidates write or follow algorithms

Question 2(a) gave candidates pseudocode (including a function) and were asked to state the output values from various lines. Most candidates were able to do this for more simple output. Fewer candidates completed the function call and return values accurately.

Many candidates struggled to differentiate between `name` (as an identifier for a variable) and `"name"` (as a string).

?	**Misconception**	`print("name")` will output the literal string "name" to the screen". `print(name)` will print the contents of the variable with identifier `name`.

Question 3(b)(i) was answered very well. However, 3(b)(ii) was only answered correctly by a small proportion of candidates. These were also the candidates who generally went on to achieve high marks on this paper.

The question asked candidates to state how many parameters are passed into the function from the line `giveChange(money - price)`. Two variables are inside the brackets. Candidates did not recognise this as a calculation.

This calculation would be completed before the function call. Only the result is passed into the function as the parameter. This means that the correct answer is one. Most candidates gave the (incorrect) answer of two.

?	**Misconception**	Where a sub program (function or procedure) has multiple parameters passed into it, there will be separated by commas – for example `testfunction(x,y)`. A subprogram with a calculation as a parameter will pass the result of this calculation into the sub program.

Question 3(c) asked candidates to draw flowchart and this was done particularly well. Many candidates changed the given pseudocode algorithm into a well-defined process that covered the same steps as the pseudocode.

A typical answer for this question would be good to use for centres when teaching about using flowcharts. Candidates who use flowcharts for algorithm answers are often not specific enough with the steps required. This gives answers that are generic or high-level. The structure of this question is a good example of the level of detail required.

Question 3(f) asked candidate to rewrite an algorithm that used `switch/case` to perform the same actions but using `IF` statements. It was clear that many candidates did not understand the use of switch/case. Some languages (such as Python) do not include support for these. It is important that these constructs are still taught.

i	**OCR support**	Appendix 5f of the GCSE Computer Science specification covers the pseudocode guide for this examination. Candidates do not need to use this in their responses but they should be aware of this as questions will be presented using this format. The current specification is available here : https://www.ocr.org.uk/images/225975-specification-accredited-gcse-computer-science-j276.pdf

Question 6(c) asked candidates to write an algorithm to:

1. Call an attendance register from a given array.
2. Count and output how many students were present and absent.

It was pleasing to see candidates decompose the problem and tackle each part to build a full solution.

The bullet points given in the question served as a scaffold.

Where one section was incorrect, other marks could be given.

A number of candidates attempted to use a `FOR…IN` loop to iterate over the given student array.

`FOR…IN` is not listed in the specification but is an entirely suitable way to approach the problem. Where this was logically sound, full marks were available.

Other candidates used FOR…NEXT loops or WHILE loops in conjunction with the length of the array. Again, marks were given where the solution was logically sound.

Marks were more limited for those candidates who did not attempt any sort of loop, but some were still able to be given where appropriate.

	AfL	Centres should be encouraged to connect together content within the specification, so that rather than teaching arrays in isolation, they could perhaps be combined with iteration to be able to count up or total values in an array, or apply this to how searching and sorting algorithms work. Where this is done on a regular basis, questions such as 6(c) perhaps become less daunting for candidates.

Other questions

It is obvious from candidates' answers that centres have a well-developed understanding of many topics covered in this paper.

Question 2(b) was answered extremely well. Many candidates were able to articulate the advantages of high-level languages over assembly language. Even more pleasing were the many candidates able to discuss issues such as portability of code and that a high-level instruction translates into many assembly instructions.

Testing (Question 3(a)(i) and 3(a)(ii)) and image representation (Question 5(a)) were similarly answered very well by many candidates.

Question 3(e) covered SQL. It was clear that many candidates were not sure what was being asked for in this question. Many incorrectly attempted to use keywords such as IF and THEN. The use of SQL in this specification is limited to a small number of keywords, as listed in the specification.

i	**OCR support**	Page 43 of the specification lists the SQL keywords that should be covered. These are limited to `SELECT`, `FROM`, `WHERE`, `LIKE`, `AND`, `OR` and wildcards. The current specification is available here : https://www.ocr.org.uk/images/225975-specification-accredited-gcse-computer-science-j276.pdf

Computer Science related mathematics questions

These were well answered for the more commonly covered areas.

Conversion to binary and hexadecimal is well covered by centres, as are simple binary shifts. Logic systems was also pleasingly well answered by the vast majority of candidates.

Fewer candidates were confident with the use of MOD and DIV (question 4di and 4dii) and some candidates tended to give answers demonstrating this, such as '3 remainder 1' for either answer.

Question 5(b) was not quite so confidently covered by many candidates. Many were able to establish that the image was made up of 90 pixels. However, the correct calculation of a suitable bit depth per colour (where two bits would be needed to represent three colours) did not appear frequently in answers.

Many candidates assumed that three colours would need three bits, or that all images used eight bits per pixel.

?	**Misconception**	With x bits per pixel, a maximum of 2^x colours can be represented. Therefore one bit can represent up to 2^1 (2) colours. Two bits would be able to represent 2^2 (2x2 = 4) colours.

Contextual questions

Candidates often ignored the context of the question in their answer.

Question 6(a) gave students an array of student names and asked for the steps taken by a linear search to find the name 'Anna' in this array.

Despite the question clearly focusing on the item to be found, many candidates instead gave a generic response outlining how a linear search could be used on any array to find any name.

This lack of detail limited the marks available to candidates.

	AfL	When looking at sample or past examination questions, it is good practice to teach students to focus on the expectations of the question; how many marks are available, what command word(s) are used in the question and whether a context or scenario is given. If context is given, answers should be provided that include this context wherever possible.

Key teaching and learning points – comments on improving performance

To improve performance across this paper as whole, teachers and students should focus on:

- Making sure sufficient programming practice takes place. The strongest candidates are confident with both writing their own algorithms and tracing through algorithms given in pseudocode.

- Making sure that candidates and teachers are familiar with the pseudocode guide given in section 5f of the specification. This will help candidates be aware of how questions will be presented to them in examinations.

- Where contexts or scenarios are given in questions, candidates should use or refer to this in their responses if appropriate.

- Make sure that the SQL keywords listed in the specification are covered with candidates.

Guidance on using this paper as a mock

When marking the answers take note of the annotations in the mark scheme.

- Marks are usually given as 1 per bullet point, unless otherwise stated.
- If a candidate gives two answers that meet the same bullet point then only 1 mark is given.
- The use of a // on a line indicates an alternative answer for that bullet point. Only 1 mark can be given even if multiple answers that hit this point are given.
- The use of a / on a line indicates an alternative word e.g. Question 2(b) easier/quicker to maintain/debug/spot errors. 'Easier to maintain' and 'quicker to debug' would both hit this mark point, but a maximum of 1 mark should be given.
- The use of bold indicates that that idea must be present, although it could be expressed in a variety of ways. For example in Question 2(c) 'Each character has a **unique** value'. The word 'unique' does not need to be present but the idea that this value is different or distinct from the values other characters are given is important.

For algorithm questions, the following guidance should be considered:

- Candidate responses do not have to be in any specified language or format. Do not mark something wrong because 'it wouldn't work in Python / VB / Java'.
- Answers must show the basic construct to use. For example, Question 6(c) requires candidates to iterate through an array. This could be a FOR loop, a WHILE loop or another sensible method of demonstrating that each element of the array is accessed. It would not be enough to simply answer 'loop around the array' – an examiner would quite rightly ask 'how?' to this.
- Structured English and flowcharts are both acceptable methods of answering algorithm questions. A candidate's answer **must** show the steps to be taken to be given marks. Too often, candidates using these methods simply repeat the question and do not show how the problem will be solved.
- Ignore any superfluous or additional code given as long as it does not contradict other parts of the answer.

Supporting you

Review of results

If any of your students' results are not as expected, you may wish to consider one of our review of results services. For full information about the options available visit the OCR website. If university places are at stake you may wish to consider priority service 2 reviews of marking which have an earlier deadline to ensure your reviews are processed in time for university applications.

Supporting you through 2020-2021

Our priority is supporting you and your students this autumn and to support you as you prepare for summer 2021 exams. We'll update our website information regularly with resources, guidance and key information.

Take a look at our support for:

- Teachers
- Students
- Exams officers
- Assessment specialists

Keep up-to-date

We are sending a weekly roundup to tell you about important updates. You can also sign up for your subject specific updates. If you haven't already, sign up here.

OCR Professional Development

Attend one of our popular CPD courses to hear directly from a senior assessor or drop in to a Q&A session. All our courses for the academic year 2020-2021 are being delivered live via an online platform, so you can attend from any location.

Please find details for all our courses on the relevant subject page on our website or visit OCR professional development.

Signed up for Exambuilder?

ExamBuilder is the question builder platform for a range of our GCSE, A Level, Cambridge Nationals, Cambridge Technicals and Functional Skills qualifications. See the full list of available qualifications in the sign up form.

ExamBuilder is **free for all OCR centres** with an Interchange account and gives you unlimited users per centre. We need an Interchange username to validate the identity of your centre's first user account for ExamBuilder.

If you do not have an Interchange account please contact your centre administrator (usually the Exams Officer) to request a username, or nominate an existing Interchange user in your department.

Need to get in touch?

If you ever have any questions about OCR qualifications or services (including administration, logistics and teaching) please feel free to get in touch with our Customer Support Centre.

General qualifications
01223 553998
general.qualifications@ocr.org.uk

Vocational qualifications
02476 851509
vocational.qualifications@ocr.org.uk

For more information visit
- **ocr.org.uk/i-want-to/find-resources/**
- **ocr.org.uk**
- **/ocrexams**
- **/ocrexams**
- **/company/ocr**
- **/ocrexams**

We really value your feedback

Click to send us an autogenerated email about this resource. Add comments if you want to.
Let us know how we can improve this resource or what else you need. Your email address will not be used or shared for any marketing purposes.

I like this **I dislike this**