Zig Zag Education

Computer Science GCSE (9–1) | OCR | J277

Practice Exams

for OCR GCSE Computer Science (J277)

Component 2: Computational thinking, algorithms and programming

Update v1.1, May 2022

zigzageducation.co.uk

POD 10919

Contents

Teacher's Introduction

This resource has been produced to help your students prepare for the examination of GCSE OCR Computer Science (J277) Component 2: *Computational Thinking, Algorithms and Programming* (for examination summer 2022 onwards). It includes four original practice question papers with supporting mark schemes.

The papers have been carefully written to cover all areas of the OCR specification, which can be seen on the J277/02 coverage grid on the next two pages. Both question papers and mark schemes are designed to be in a similar style to the OCR specimen assessment materials, to provide realistic exam practice. The papers can be used as a 'mock' examinations; alternatively, individual questions can be set for recap/revision purposes.

The mark schemes include clear, bullet point answers and marking guidance, along with assessment objective (AO) mapping for every question.

Remember!

Always check the exam board website for new information, including changes to the specification and sample assessment material.

Update v1.1, May 2022
Paper B, question 2: Corrected flowchart to start with y=9999.
Paper B, question 2a: Corrected '13' to '1' and '3'.
Paper C, question 3a: clarified question; corrected answer to second line.
Paper C, question 4a(i) mark scheme: corrected answer to '/ Division'.
Paper C, question 4a(ii) mark scheme: removed '/ Division', added '– Subtraction' and '+ Addition'.
Paper C, question 5: altered code for clarity; added alternative answer to mark scheme.

OCR Specification Coverage (J277/02)

	Paper A	Paper B	Paper C	Paper D
2.1.1 Computational thinking				
Principles of computational thinking: Abstraction; Decomposition; Algorithmic thinking	✓		✓	✓
2.1.2 Designing, creating and refining algorithms				
Identify the inputs, processes, and outputs for a problem	✓			
Structure diagrams				✓
Create, interpret, correct, complete and refine algorithms using: Pseudocode; Flowcharts; Reference language/high-level programming language	✓			✓
Identify common errors			✓	
Trace tables	✓			✓
Standard searching algorithms: Binary search; Linear search	✓ (Linear search)	✓ (Binary search)		
Standard sorting algorithms: Bubble sort; Merge Sort; Insertion sort	✓ (Merge sort)		✓ (Bubble sort)	✓ (Insertion sort)
2.2.1 Programming fundamentals				
The use of variables, constants, operators, inputs, outputs and assignments	✓	✓	✓	✓
The use of the three basic programming constructs used to control the flow of a program: Sequence; Selection; Iteration	✓	✓	✓	✓
The common arithmetic operators	✓	✓	✓	✓
The common Boolean operators AND, OR and NOT	✓	✓		

	Paper A	Paper B	Paper C	Paper D
2.2.3 Additional programming techniques (cont.)				
How to use sub programs (functions and procedures) to produce structured code			✓	✓
Random number generat				
2.3.1 Defensive design				
Defensive design considerati Anticipating misuse; Authentication				✓
Input validation	✓			✓
Maintainability: Use of sub progra Naming conventions; Indentation; Commenting		✓		✓
2.3.2 Testing				
The purpose of testing				✓
Types of testing: Iterative; Final/Term a				✓
Identify syntax and logic errors	✓			✓
Selecting and using suitable test data: No mal, Boundary, Invalid/Erroneous	✓			
Refining algorithms			✓	✓
2.4 Boolean Logic				
Simple logic diagrams using the operators AN OR and NOT		✓	✓	
Truth tables	✓	✓	✓	
Combining Boolean operators using AND, OR and NOT		✓	✓	
Applying logical operators in truth tables to solve problems		✓		
2.5.1 Languages				
Characteristics and purpose of different levels of programming language:				

OCR GCSE (9–1) Computer

J277/02 Computational thin
algorithms and programm

Practice Exam (Question Paper A

Name	

Time allowed: 1 hour 30 minutes

Instructions:

- Use black ink
- Write your name at the top of this p
- Answer **all** questions in the spaces provided

Information:

- The maximum mark for this paper is 80
- The marks for each question are shown in brackets.

Answer **all** the questions.

Section A

1 Tick (✓) **one or more** boxes in each row to identify whether each stat
constant, or both.

Statement	Variable
It can be changed when the program is not run	
It has an identifier	
It cannot be changed while the program is running	
It is a named memory location	

2 A program is written to calculate the average (mean) temperature and
24 hours.

(a) Identify **two** inputs and **two** outputs that will be needed in the pro

Input 1 .....

Input 2 .....

Output 1 .....

Output 2 .....

(b) 48 temperature values are stored in the one-dimensional array `t`

(i) Give **two** features of a one-dimensional array.

1 .....

.....

2 .....

.....

(ii) Explain why an array is a more appropriate data structure for values than the use of variables.

.....

.....

.....

.....

.....

.....

.....

.....

(iii) Complete the pseudocode program to calculate the average temperature.

```
total = .....
for x = 0 to 47
  total = ..... + tempera
next x
average = total ..... 48
print("The average temperature is " +
```

(c) The function `even` counts and returns the quantity of numbers in

Complete the pseudocode function `even`.

```
function even(temperatu
  count = 0
  for x      47
    if temperature[x] .....  2 == 0 th
      count = count .....
    endif
  next x
  return .....
endfunction
```

3 The following pseudocode program outputs the times table for a numb to 12. For example, if the user enters 5, it will output:

5	10	15	20	25	30	35	40	45	50	55	60

```
number = input("Enter a number")
for x = 1 to 12
  print(number * x)
next x
```

(a) State how you would ma[illegible] algorithm to output the times tab to 14.

.....

.....

(b) Rewrite the pseudocode program using a condition-controlled loo

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

4 **Fig. 1** shows the data stored in a program.

Horse	Cat	Hamster	Frog	Lizard	Whale	D

Fig. 1

(a) Show the stages of a merge sort when applied to the data shown

.....

.....

.....

.....

.....

.....

.....

.....

(b) The data needs to be searched to identify whether the data 'Whal

(i) Identify **one** searching algorithm that can be used to search f **Fig. 1**.

.....

(ii) Show the stages of the search algorithm you identified in **par** 'Whale'.

.....

.....

.....

.....

.....

.....

.....

.....

5 Five students are working together to create a computer game for a s
allow a user to drive a car around a virtual environment.

(a) The students use principles of computational thinking when desig

Complete the table by defining each principle of computational th
of how each can be applied to this computer game.

Principle	Description	Applica
Abstraction		
L position		

(b) The programmers use a high-level language to write the program

Give **two** characteristics of high-level languages.

1 .....

.....

2 .....

.....

(c) The programmers choose to use an Integrated Development Env
the program.

Describe the following tools and facilities available in an IDE.

Editor .....

.....

.....

.....

Translator .....

.....

.....

.....

6 Complete the truth table in **Fig. 2** for the logic statement: `X = (A A`

A	B	C	X
0	0	0	
0	0	1	1
0	1	0	0
0	1	1	
1	0	0	0
1	0	1	1
1	1	0	
1	1	1	

Fig. 2

7 The database table `Logins` stores the time that each employee logs

EmployeeID	Date	Time
AW1	02/02/2021	10:02
DG67	02/02/2021	10:15
LL3	02/02/2021	9:58
AW1	03/02/2021	8:37
DY7	03/02/2021	12:29
LL3	03/02/2021	9:04
HH9	03/02/2021	10:12

Complete the SQL query to return the Employee ID of all employees w
2nd February 2021.

```
SELECT .....

FROM .....

WHERE .....
```

Section B

We advise you to spend at least 40 minutes on this

Some questions require you to respond using either the OCR Exa
a high-level programming language you have studied. These

8 A programmer is designing a new computer game. The user will battle
computer-controlled character.

Each character has three values:

1. Character name, e.g. BattleCat, Jane, Joe
2. Current health, e.g. 1, 50, 10
3. Defence strength, e.g. 0.1, 0.5, 1.0

(a) Tick **one** box in each row to identify the most appropriate data

Data	Boolean	Integer	Real	
Name				
Health				
Defence				

(b) The current health is randomly generated as a whole number bet

Write a program statement to randomly generate a value for heal
named `health`.

You must use **either**:

- OCR Exam Reference Language, **or**
- a high-level programming language that you have studied

.....

.....

.....

(c) On a user's turn to attack, they have to answer a question. They
a question with a level of difficulty of low ('L'), medium ('M') or hig

When a valid level of difficulty is entered, the procedure `questi`
is called.

(i) Complete the following program to:

- take the question choice as input continually until valid
- call the procedure question with the level of difficulty as

The code `variable.upper` returns the characters in `var`

You must use **either:**

- OCR Exam R... Language, **or**
  a high ... gramming language that you have studi

```
levelOfDifficulty = ..........("Enter L
  M for medium, H for high")

levelOfDifficulty = ..........

..... levelOfDifficulty = "L" .....

levelOfDifficulty = "M" ..... levelOfDif

question(levelOfDifficulty)
```

(ii) Complete the table by identifying the test type and expected
items of test data for the program in **8(c)(i)**.

Test data	Test type	E
"L"		
"m"		
0		
"high"		

(d) At the moment the program is tested using only one low-, one me
question. The questions are mathematical questions, e.g. '3 + 2'.

The program stores the questions in three global variables: `lowQu`
and `highQuestion`. The program stores the answers in three
`lowAnswer`, `mediumAnswer` and `highAnswer`.

The procedure `question()`:

- takes the level of difficulty as a question
- outputs the question for that difficulty
- takes the answer as input from the use
- outputs whether the answer is correct or incorrect

```
01  procedure q  stion(levelOfDifficulty.upp
02    i  levelOfDifficulty == "L" then
        answer = input(lowQuestion)
        if answer == lowAnswer then
          print(/Correct/)
        else
          print("Incorrect")
        endif
      elseif levelOfDifficulty < "M" then
        answer = input(mediumQuestion)
        if answer == mediumAnswer then
          print("Correct")
        else
          print("Incorrect")
        endif
      elseif
        answer = input(highQuestion)
        if answer == highAnswer then
          print("Correct")
        else
          print("Correct")
        endif
      endif
    endprocedure
```

(i) There are **four** errors in the code. Identify the line number and c

Line number .....

Corrected cod .....

e n mber .....

Corrected code .....

Line number .....

Corrected code .....

Line number .....

Corrected code .....

(e) A set of low-level questions and answers is stored in the text file 1
program needs to store the questions and answers in a 2D array o

(i) Write a program to declare a 2D array with space for 100 que

You must use **either:**

- OCR Exam Reference Language, **or**
- a high-level programming language that you have stu

.....

.....

.....

(ii) Data in a file called `lowLevelFile.txt` is presented as f

Question 1
Answer 1
Question 2
Answer 2
...

The file stores an unknown number of questions up to a max

Write a program to:

- open the file `lowLevelFile.txt` and read all of into the 2D array you declared in **part 8 (g)(i)**
- output the number of questions read from the file

You must use **either:**

- OCR Exam Reference Language, **or**
- a high-level programming language that you have s

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

(f) The following program calculates the number of points a user gai

```
function total(low, medium, high)
  lowPoints = low
  mediumPoints = medium * 5
  highPoints = high * 10
  total = lowPoints + mediumPoints + hig
  return total
endfunction
```

Complete the trace tab[illegible] this program when the following p

```
[illegible]0, 5, 8)
```

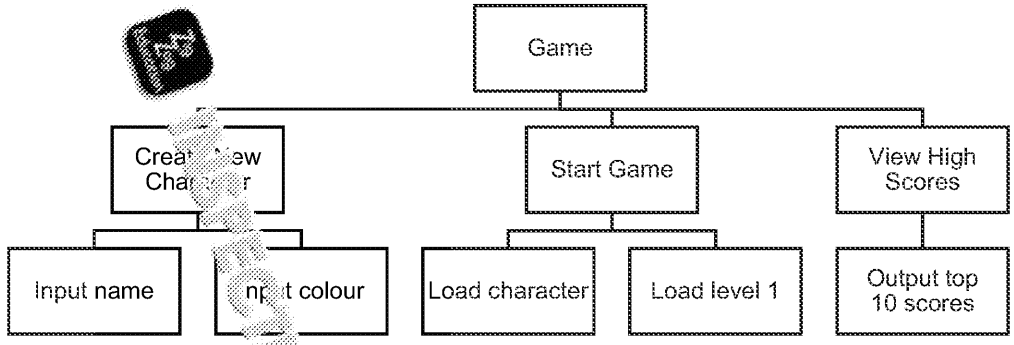
lowPoints	mediumPoints	highPoints	t

Preview of Questions Ends Here

Practice Paper D – Mark Scheme

Question	Answer	Marks	Guidance
1a	• a • b • c	2 (AO2)	1 mark each to max. 2 marks
1b(i)	• To stop his program being released with errors • To stop users being unable to use the program • So he can make sure it works as intended • So he can make sure all routes through the program work • So he can make sure it cannot be broken • So he can make sure any/all errors are caught and handled	2 (AO2)	1 mark each to max. 2 marks
1b(ii)	• Iterative – Karl needs to test the program during the development to make sure each part of the program works before moving on • Final/terminal testing – Karl needs to test at the end to make sure that the entire program works prior to release	2 (AO2)	1 mark for each; answers are examples only
1b(iii)	• Initialising a and b to 0 • Column a going from 1 to 3 • Column c taking all inputs in order • Column b correctly adding to the total • Output being the total in b / 14 with follow-through for incorrect addition in b	5 (AO2) (AO3)	1 mark for each point

a	b	c	OUTPUT
0	0	1	
1	1	3	
2	4	10	
3	14	20	

Question	Answer	Marks	Guidance
2a	e.g.  	5 (AO3)	1 mark for a suitable diagram that includes the following: • Structure diagram format, i.e. boxes with lines connecting • Level 1 including all three menu options • Level 2 below 'Create new character' includes input name and colour • Level 2 below 'Start game' includes load character and load level 1 • Level 2 below 'View high score table' includes output top 10 scores Ignore any additional boxes. Allow multiple items per box, e.g. Input name & Input colour
2b	Abstraction • Remove unnecessary features/components • Focus on the required features/components • Reduce the memory requirements for the program Algorithmic thinking • Identifying the steps in a program • Writing the steps in a form that a computer can process	3 (AO1)	• 1 mark for name • Max 2 for description
3a	• Make Banana the sorted list • Put Apple in sorted list before Banana • Leave Peach, Pear and Strawberry in place • Move Grapes • Move Raspberry	6 (AO2)	1 mark for description of each point Accept a description of visual application

<table>
<tr><th>Question</th><th>Answer</th><th>Marks</th><th>Guidance</th></tr>
<tr><td>3b</td><td>• It checks whether the value is less than or greater than the central value // it splits the list into those less than, and those greater than, the value
• There may be values greater than the value in the less than sub-list</td><td>1
(AO1)</td><td>• 1 mark each max. 2 marks
• Accept any suitable description</td></tr>
<tr><td>4a</td><td><table><tr><th>Term</th><th>Description</th></tr><tr><td>Anticipating misuse</td><td>• Consider how users may use the system incorrectly
• … put in place features to handle these / stop the program crashing</td></tr><tr><td>Authentication</td><td>• Introduce features such as passwords / two-step authentication
• … so that only authorised users can access the system</td></tr></table></td><td>(AO1)</td><td>• 1 mark for each point
• Max. 2 marks for each description</td></tr>
<tr><td>4b</td><td>• Range check
• … make sure the number is only between 100 and 200 inclusive
• Presence check
• … make sure the user has entered a number
• Type check
• … make sure the number entered is a whole number / does not include other characters / decimals</td><td>3
(AO2)</td><td>• 1 mark each
• Max. 2 marks for only naming checks without application</td></tr>
<tr><td>4c</td><td><table><tr><th>Feature</th><th>How it improves maintainability</th></tr><tr><td>Commenting</td><td>• Explains each section of code
• When you return you can understand what each part of the program does
• Other people who need to amend the program can understand what each part of the program does</td></tr><tr><td>Indenting</td><td>• Shows which code is contained within each construct
• Allows you to easily edit the code within one construct
• … without having to work out which code is contained within it</td></tr></table></td><td>6
(AO1)</td><td>• 1 mark per bullet
• Max. 2 marks for each feature
• Points are examples only; other suitable explanations can be awarded marks</td></tr>
</table>

Question	Answer	Marks	Guidance

5	• Switch on value1
	• Case for 0 and output
	• Cases for 1, 2, and outputs
	• Case default and output
	• Remainder of the program correct
	
	e.g.
	`value1 = input("Enter a number")`
	`value2 = input("Enter a number")`
	`value3 = input("Enter a number")`
	`switch value1:`
	`  case 0: print(value2 + value3)`
	`  case 1: print(value2 - value3)`
	`  case 2: print(value2 * value3)`
	`  default: print(value2 / value3)`
	`endswitch`
		5
		(AO3)
			1 mark for each point

6a	• Each card has two values to store • … that can be represented as a table • Each card is in its own row in the table	2 (AO2)	1 mark for each point to max. 2 marks

6b	`filename = "cards.txt"`
	`file = open(filename, r)`
	`for(count = 0 to 51)`
	`  print("Number " + myFile.readLine())`
	`  print("Shape " + myFile.readLine())`
	`next count`
	`myFile.close()`
		2
		(AO3)
			1 mark for each statement

Question	Answer	Marks	Guidance
7a(ii)	e.g. `Player numb 3 has scored 0 and played 2 games`	3 (AO3)	1 mark for each point: • Output of 0 • Output of 2 • Order and remainder of text correct

7b	e.g.
	`procedure gam sult()`
	`  repeat`
	`  won = input(Enter the winning player's number")`
	`  until won >= nd won <= 9`
	`  repeat`
	`  lost = input(Enter the losing player's number ")`
	`  until lost >= 0 d lost <= 9`
	`  players[won][0] = layers[won][0] + 1`
	`  players[won][1] = layers[won][1] + 1`
	`  players[lost][1] = players[lost][1] + 1`
	`endprocedure`
		5
		(AO3)
			High-level programming language / Exam Reference Language response required
			
			Do not accept pseudocode / natural English
			
			Identifiers must not have speech marks
			Strings and outputs must have speech marks
			
			1 mark for each completed statement
			
			Accept array elements combined, e.g. `[won,1]`

7c(i)	• Line 02 `winner = -1` (winner needs to be below all possible points) • Line 05 `if players[x][0] > winningPoints then` (< should be >) • Line 07 `winner = x` (assignment statement must be the other way around) • Line 09 `return winner` (`winner` should be returned, not `x`)	4 (AO3)	**High-level programming language / Exam Reference Language response required** Do not accept pseudocode / natural English Identifiers must not have speech marks Strings and outputs must have speech marks 1 mark for each error and correction Corrections are examples only; accept other valid corrections

Question	Answer	Marks	Guidance

7d	• Procedure header with correct identifier
	• Loop through all 10 players
	• Output p… number in loop
	• Output pl… ore in loop
	• Appropriate output
	
	e.g.
	`procedure outp…ll()`
	`  for x = 0 to …`
	`  print("Playe…" + x + " has scored " + players[x][0]`
	`  next x`
	`endprocedure`
		5
		(AO3)
			High-level programming language / Exam Reference Language response required
			
			Do not accept pseudocode / natural English
			
			Identifiers must not have speech marks
			Strings and outputs must have speech marks
			
			…nark for each point

7e	• Output a menu and …g a value from the user
	• … validating it is appr…iate
	• Check whether inputting result is selected…
	• … call `gameResult()`
	• Check winner is selected…
	• … call `winner()`…
	• … output return value
	• Check if all scores want to be output…
	• … call `outputAll()`
	
	e.g.
	`choice = 0`
	`while choice <> 1 and choice <> 2 and choice <> 3`
	`  choice = input("Enter 1 to enter the result of a match, enter 2 to`
		7
		(AO3)
			Hig…vel programming language / Exam Refe…ce Language response required
			
			Do not …ept pseudocode / natural English
			
			Identifie…ust not have speech marks
			Strings a…utputs must have speech marks
			
			1 mark for each point to max. 7 marks

Preview of Answers Ends Here