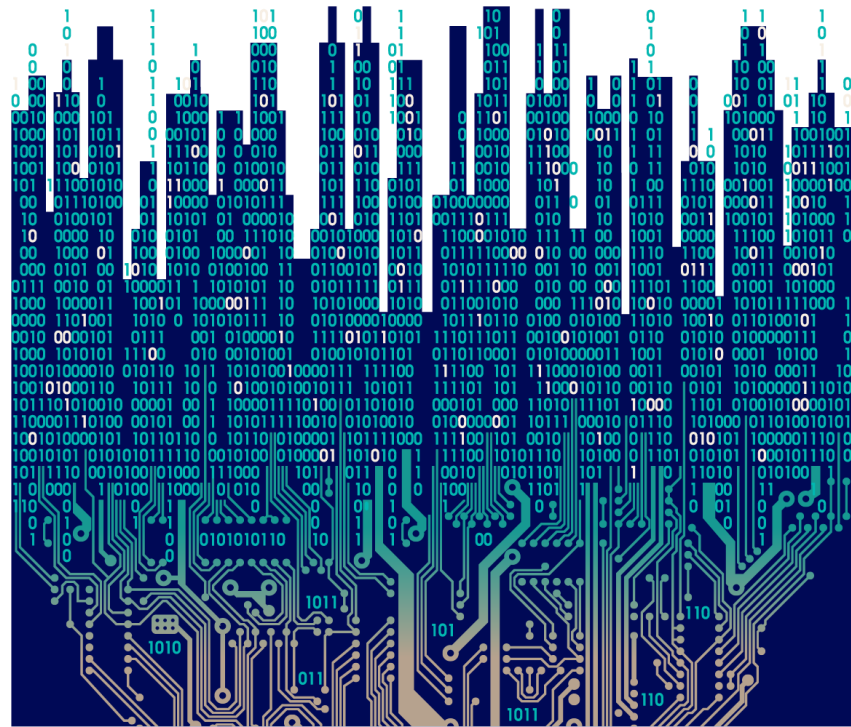


# GCSE Computer Science

Preparing for the exams



# Agenda

- Section 1 – Component requirements
- Section 2 – Paper 1
- Section 3 – Paper 2
- Section 4 – Logistics for Paper 2
- Section 5 – Other resources
- Section 6 – Q&A

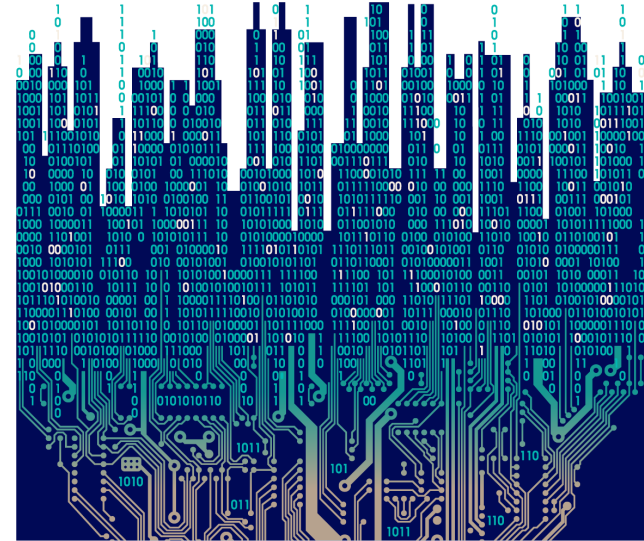
# Question 1

Let's find out a little about our group today.

**Answer in the chat window:**

1. What are the key reasons for attending the session today?
2. What is the single most important thing you hope to take away from the session?

# Section 1 – Component requirements



# Qualification breakdown

## **Paper 1: Principles of Computer Science (\*Paper code: 1CP2/01)**

***Written examination: 1 hour and 30 minutes***

***50% of the qualification***

***75 marks***

### **Content overview**

This paper will assess Topics 1 to 5.

## **Paper 2: Application of Computational Thinking (\*Paper code: 1CP2/02)**

***Onscreen examination: 2 hours***

***50% of the qualification***

***75 marks***

### **Content overview**

This paper will assess Topic 6: Problem solving with programming.

# Assessment objectives

Students must:		% in GCSE
<b>AO1</b>	Demonstrate knowledge and understanding of the key concepts and principles of computer science	30
<b>AO2</b>	Apply knowledge and understanding of key concepts and principles of computer science	40
<b>AO3</b>	Analyse problems in computational terms: <ul style="list-style-type: none"><li>• to make reasoned judgements</li><li>• to design, program, evaluate and refine solutions.</li></ul>	30
<b>Total</b>		<b>100</b>

# Breakdown by paper

Paper	Assessment Objectives			Total % for all Assessment Objectives
	AO1 %	AO2 %	AO3 %	
Paper 1: Principles of Computer Science	30	20	0	50
Paper 2: Application of Computational Thinking	0	20	30	50
<b>Total for GCSE</b>	<b>30</b>	<b>40</b>	<b>30</b>	<b>100</b>

## Question 2

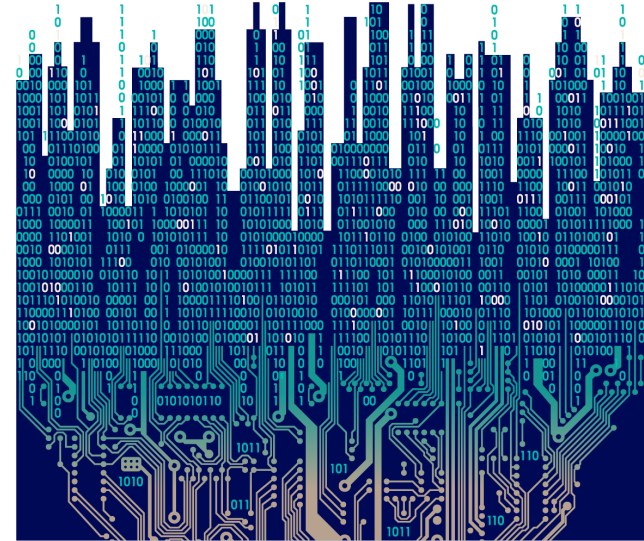
Let's see how experienced you are with this qualification.

### Chat window:

1. Type 'New' if you've not been through a live exam series.
2. Type how many live exam series you've done with this qualification.



# Section 2 – Paper 1



# Structure

- The paper requires students to demonstrate and apply knowledge and understanding of the key principles and concepts
- This is an un-tiered paper
- The paper consists of five questions
- Each question focuses on a different topic from the specification
- The order of topics changes each year
- The last sub-question in question four is a discussion question
- The last sub-question on the paper is a higher-tariff question and is not a discussion question.

**1 Computers**

(a) The CPU contains a number of components.

(i) Complete the table with the correct bus for each role. (3)

Bus	Role

memory to the CPU.

(ii) Identify the component inside the CPU that stores data. (1)

☒ **A** Arithmetic logic unit

☒ **B** Clock

☒ **C** Main memory

☒ **D** Register

# Command word taxonomy

- Command words are used consistently in the paper to indicate the type of response expected.
- All command words are set out in the specification document on page 27.

Command word	Mark Tariff	Definition
Amend	1 to 6 marks	Requires changes or additions to code, or deletions or rearrangement of code.
Calculate	2 to 4 marks	Obtain a numerical answer, showing relevant working. If the answer has a unit, this must be included.
Complete	2 to 6 marks	Requires the completion of a table/diagram/algorithm (in any notation).
Construct	2 to 4 marks	Usually requires creation of an artefact using subject-specific symbolic representation, rules and syntax.
Convert	2 to 6 marks	Requires changing information from one symbolic representation to another representation. May require amending to provide new functionality/facility.

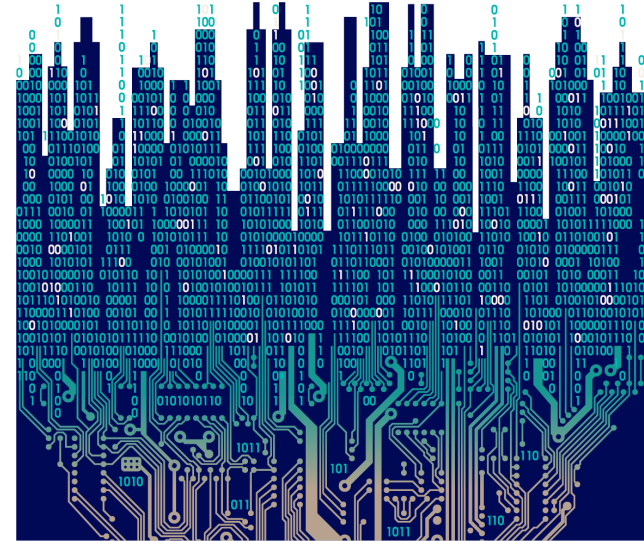
- The 'Getting Started Guide' is essential reading alongside the specification.

# Question types

- Multiple-choice
  - Uses the command word 'Identify'
- Short open
  - Usually requires one or a few words
- Medium open
  - May require a few sentences
- Extended open
  - May use the command word 'Discuss'
  - Usually the last sub-question in question four
- Binary conversion
  - Requires creating an expression or completing subject-specific mathematics
- Completing tables/diagrams
  - Can be used anywhere in the paper
- Non-essay high-tariff
  - Usually the last question or sub-question in question five, the last on the paper

# Section 2 – Paper 1:

## Two-mark Explain



# Marked example

(g) Some users are given administrator privileges.

2022 Q02g

Explain **one** way an operating system allows an administrator to manage users.

(2)

Question Number	Answer
2(g)	<p>A linked explanation such as:</p> <ul style="list-style-type: none"><li>• Users can be added/deleted (1) so multiple people can use the same computer (1)</li><li>• Edit user permissions (1) so only specific users can securely access their storage space (1)</li><li>• Control the amount of resources/storage each user can access (1) so the limited storage on the machine can be shared (1)</li><li>• Enforce user permissions (1) so only certain users are allowed to install programs / access certain files (1)</li></ul>

Administrators can give or deny users access to files.  
This means that an administrator can decide <sup>what files</sup> ~~whether~~ a  
user logged into the system can view or edit

2 marks

Administrators can decide what files a user logged into the system can view or edit (1) **because** they can give or deny users access to files (1)

Model Answer

# In the classroom

- Create practice questions and mark schemes across topics

2022 Q01e

- Teachers or students.

- Practice answering the questions

- Make a statement
  - Use a connective word (because) to show reasoning or justification
  - Provide an expansion.

(e) Explain **one** disadvantage of using a star network topology.

Question Number	Answer
1(e)	<p>A linked explanation such as:</p> <ul style="list-style-type: none"><li>• All communication could fail (1) because/if the central device fails (1)</li><li>• It can be difficult/expensive to set up (1) because each device needs a cable to connect to the central device (1)</li><li>• The number of devices that can be connected is limited (1) because the central device supports a fixed number of connections (1)</li></ul>

- Practice the response format

- Give a list of statements relevant to a question
  - Make a 2-mark response, using because
  - Remember, the expansion must work logically with the statement.

# In the exam

- Read the question carefully, underline command words, keywords, or terms
- Recall several relevant facts in the context of the question
- Identify a statement and an expansion from the recalled information
- Make a 2-mark response, using the connecting word **because** to link the statement and the expansion
- Remember, the expansion must work logically with the statement, to show an element of reasoning or justification. Two statements cannot be awarded both marks.
- Do **not** repeat the information provided in the question.

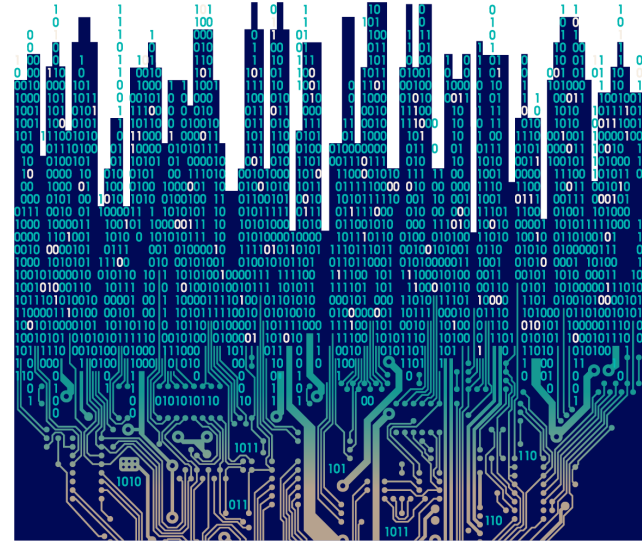


© gustavofrazao / 123RF



# Section 2 – Paper 1:

## Multiple-mark Describe



# Marked example

(c) Describe how a firewall protects a local area network (LAN).

2022 Q01c

(2)

Question Number	Answer	
1(c)	<p>A description to include two from:</p> <ul style="list-style-type: none"><li>• Monitors/checks traffic (1) using a set of rules / list of authorised/unauthorised computers/addresses/protocols (1) to decide if data is allowed into or out of the network (1)</li></ul>	A

A Firewall protects a LAN within a WAN such as the internet by controlling communication and network traffic between both networks. The Firewall uses a pre-defined set of rules to decide what to allow from one side of the Firewall to the other.

2 marks

# In the classroom

- Create practice questions and mark schemes across topics

2023 Q04d

- Teachers or students.

- Practice answering the questions

- Make one statement
  - Make the required number of statements
  - Ensure the order is correct.

(d) A linear search algorithm can be used on both a sorted and an unsorted array.

Describe how a linear search algorithm operates on an **unsorted** array.

(4)

Question Number	Answer
4(d)	<p>Award up to <b>four</b> marks for a linked description, such as:</p> <ul style="list-style-type: none"><li>• Start at the first position / Iterate/Traverse (through the array) (1), compare the item with the target (1), stop when the target is matched (1), or stop when the end of the list is reached (and the item is not matched) (1)</li></ul>

- Practice the response format
  - Give a list of statements relevant to a question
  - Make a multiple-mark response, ensuring the order is correct
  - Remember, the statements must work logically together.

# In the exam

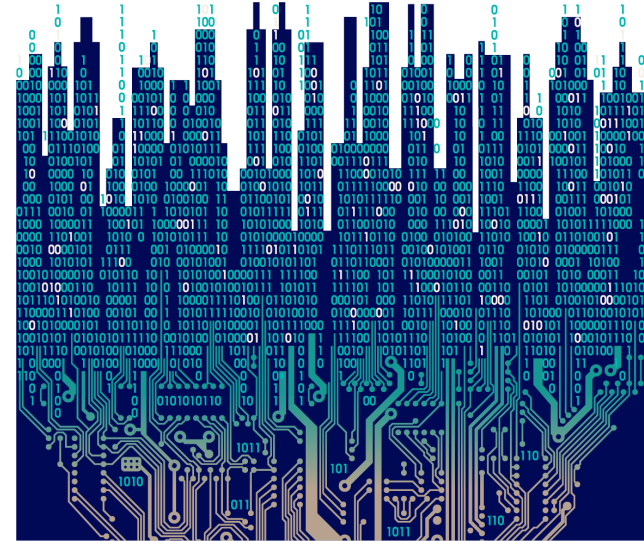
- Read the question carefully, underline command words, keywords, or terms
- Identify the number of statements required for all the marks
- Recall several relevant facts in the context of the question
- Identify facts that work together or come in a particular order
- Make a multiple-mark response, ensuring the correct order
- Remember, the statements must work logically together
- Do **not** repeat the information provided in the question.



© Petr Vaclavek / Shutterstock

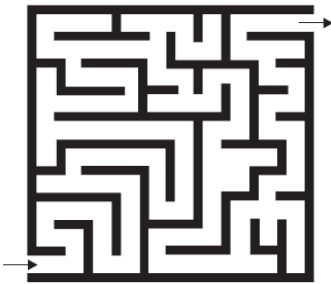
# Section 2 – Paper 1:

## Six-mark Discuss



# Marked example

- (f) A group of students are working together on a single maze game. The arrow keys control the character. When the character reaches the end of the maze without touching a wall, a happy sound is played. The game also displays a score.



Discuss the use of decomposition and abstraction in developing this game.

Your answer should include:

- a definition of each term
- the benefits each brings to the group of students
- an example of where each could appear in the program code.

(6)

Question Number	Answer
4(f)	<p>Indicative content:</p> <p><b>Definition</b></p> <ul style="list-style-type: none"><li>• Decomposition is breaking down into smaller parts. Problems, solutions, and algorithms can be decomposed.</li><li>• Abstraction is the process of removing or hiding unnecessary detail.</li></ul> <p><b>Benefits</b></p> <ul style="list-style-type: none"><li>• It is usually easier to solve several smaller problems, such as checking if touching a wall or updating the score display, than solve one big problem, such as making a game.</li><li>• Different parts of the program can be shared between the class members to speed up development, for example, one group could work on the code to control the character, while another works on creating and playing the sounds.</li><li>• Once all the pieces, like sounds, movement, and score are working correctly, the smaller solutions can be combined to make a larger solution, with fewer errors.</li><li>• The individual parts of the program, such as updating the score can be ignored by the group of students writing the code for moving the character with the arrows / allowing each group of students to focus only on the small problem they have been given means time is not wasted on analysis not relevant to the solution.</li></ul> <p><b>Appear in program code</b></p> <ul style="list-style-type: none"><li>• Different blocks of code logic show decomposition, such as</li></ul>

2023 Q04f

6 marks

Decomposition means breaking down a problem into more manageable sub problems in order to solve it. Abstraction is about focusing on the important details and leaving out specific details. Decomposition means problems become more manageable and therefore releases stress.

# In the classroom

- Teachers may create sample responses, across a range of marks, for discussion questions in SAMs, Specimens, and past papers

- Ask students to mark the responses and justify the marks given
- Ask students to improve the responses.

SAM Q04d

- Students may create sample questions and mark schemes
  - Exchange questions and answer
  - Return responses and mark, adding feedback for improvement.

- (e) A team of programmers is creating the code for an alarm system. The system uses a high-level programming language for the touchscreen graphical user interface and a low-level language for the control unit that monitors the sensors and triggers the alarm.

Discuss the characteristics of high-level languages and low-level languages that make them appropriate for the team of programmers to code these uses.

Your answer should consider:

- the purpose of the system
- the advantages of high-level languages
- the advantages of low-level languages.

(6)

Question number	Indicative content	Mark
4(e)	<p>Advantages of high-level languages:</p> <ul style="list-style-type: none"><li>High-level languages come with libraries of ready-made graphical user interface components (buttons, icons and menus), which the team can use to reduce the amount of code they have to write from scratch.</li><li>High-level languages have a range of integrated development tools, editors and syntax checkers, which will enable the team to develop the interface code more efficiently.</li><li>Portability is a real consideration: should the company decide to use a different chipset in the future, programs written in a high-level language won't need to be rewritten. They can be recompiled to run on new architecture.</li></ul>	(6)

# In the exam

- Read the question carefully, underline command words, keywords, or terms.
- Responses must be in the context of the question
- Where bullet points are given in the question paper, they should be addressed
- This is the question that uses levels-based mark schemes. Discussion must be logical, coherent, and accurate
- It is best to outline, then write the discussion so that it remains organised
- There is always a blank page somewhere in the question paper that can be used for notes or outlining. Remember, that page will not be marked
- Do **not** repeat the information provided in the question.

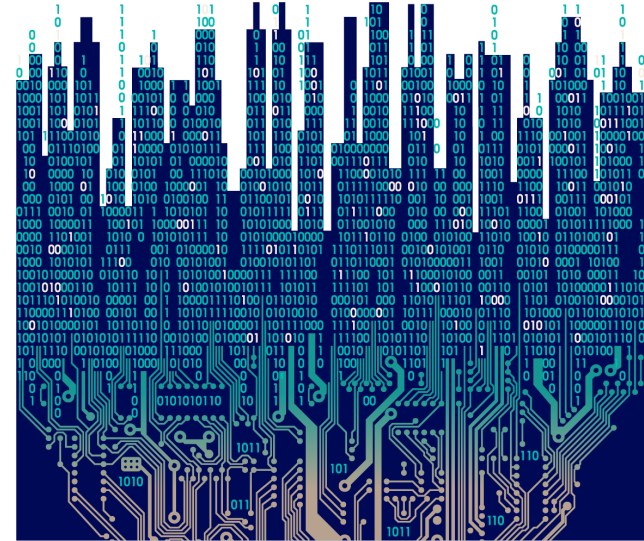


© Feng Yu / Shutterstock



# Section 2 – Paper 1:

## Six-mark non-essay



# Marked example

2023 Q05f

(f) An analogue sound is represented in digital form.

The sound is one second long and is sampled at 10Hz.

The digital representation has a bit depth of 5 and is stored in two's complement.

**Sound data:**

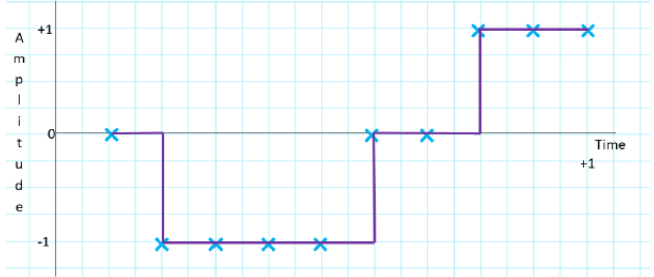
00000 11111 11111 11111 11111

00000 00000 00001 00001 00001

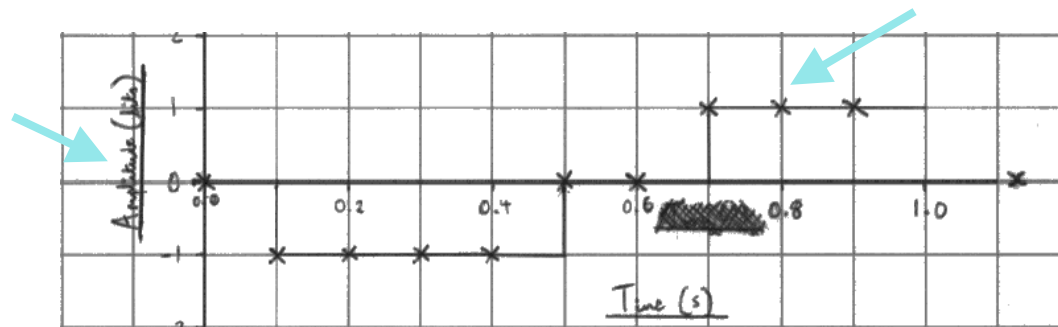
Draw a graph to represent the data sampled.

You must include:

- labels for the x and y axes
- values for the x and y axes
- each sample plotted as an X
- samples joined up to show the digital form.

Question Number	Answer
5(f)	<p>Award <b>one</b> mark for any of the following up to a maximum of <b>six</b> marks:</p> <ul style="list-style-type: none"> <li>• x-axis labelled correctly as time/seconds (1)</li> <li>• y-axis labelled correctly as amplitude/value/sample (1)</li> <li>• value labels on x-axis as 0 and 1 (1)</li> <li>• value labels y-axis as -1 and 1 (1)</li> <li>• all 10 values plotted to correct points (1)</li> <li>• points joined to form a square wave, even if not all points are there or some are plotted inaccurately (1)</li> </ul> 

(6)



6 marks

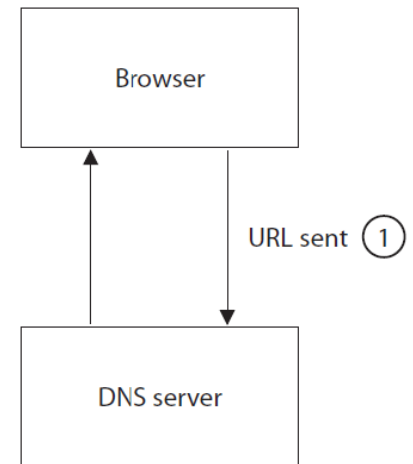
# In the classroom

- Create practice questions and mark schemes across topics
  - Teachers or students.
- Students practice answering
  - Exchange answers
  - Mark answers, adding feedback for improvement.

## Specimen 1 Q05e

- (e) Complete the diagram to show the processes used to find the IP address of a web server and download a page.

Include labels, arrows and numbers in your diagram to show the order in which processes are carried out.

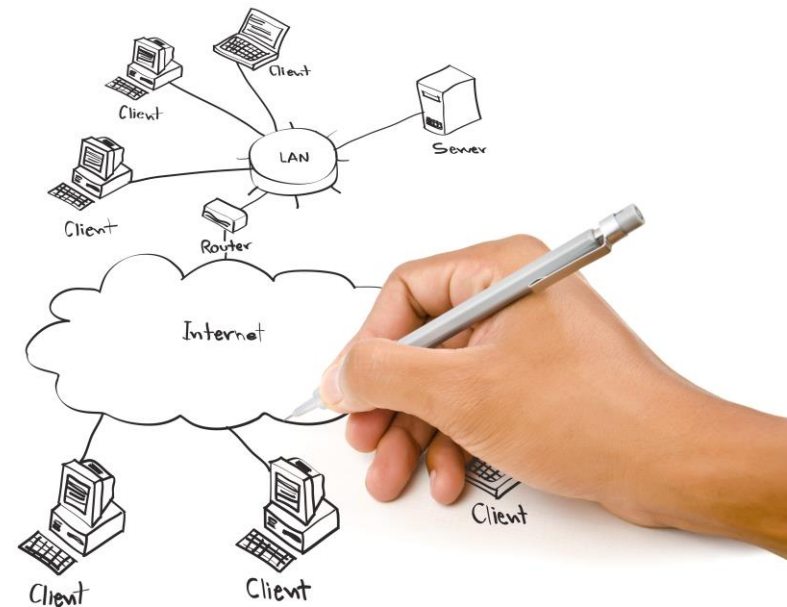


(6)



# In the exam

- Read the question carefully, underline command words, keywords, or terms
- Take note of additional instructions, like the need for labels, numbering, or items that **must** be included
- Where bullet points are given in the question paper, they should be addressed
- It is best to make a rough sketch, then do a good drawing so the final response is neat, tidy, and can be read
- There is always a blank page somewhere in the question paper that can be used for rough drawing. Remember, that page will not be marked.



© Ohmega1982 / Shutterstock

# Take away

- Classroom
  - Use published materials as part of regular teaching
  - Run a mock, under timed conditions, using one of the published past papers.
- Exam
  - Understand the requirements of the command word taxonomy
  - Provide detailed responses, including examples and reasons
  - Use subject-specific language to avoid generic terms
  - Add full brackets to arithmetic and logical expressions
  - Address bullets in questions, if provided
  - Use blank pages in paper for outlines and rough drawings.

Please check the examination details below before entering your candidate information

Candidate surname					Other names				
Centre Number					Candidate Number				
<input type="text"/> <input type="text"/> <input type="text"/> <input type="text"/> <input type="text"/>					<input type="text"/> <input type="text"/> <input type="text"/> <input type="text"/> <input type="text"/>				
<b>Pearson Edexcel Level 1/Level 2 GCSE (9–1)</b>									
Time 1 hour 30 minutes					Paper reference		<b>1CP2/01</b>		
<b>Computer Science</b>									
<b>PAPER 1: Principles of Computer Science</b>									
You do not need any other materials.								Total Marks	
<input type="text"/>								<input type="text"/>	

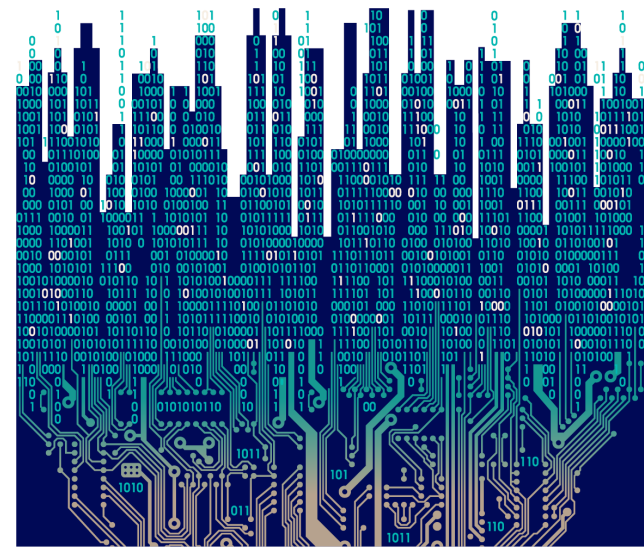
## **Question 3**

Let's see how we can help each other.

**Chat window:**

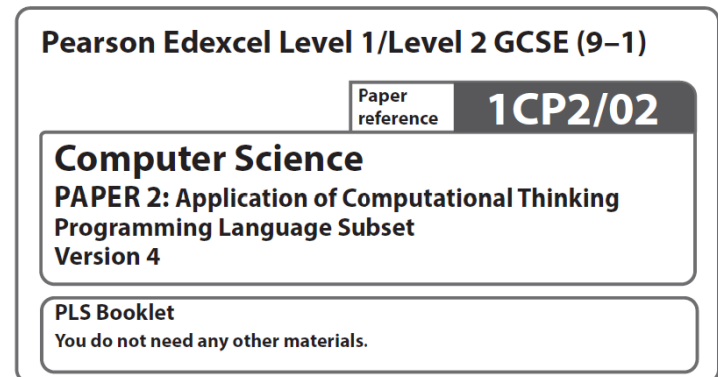
1. What methods do you use to prepare for paper 1?

# Section 3 – Paper 2



# Structure – Paper

- The paper requires students to demonstrate and apply knowledge and understanding of programming
- This is an un-tiered paper
- The paper consists of six questions
- Each question has 5 to 15 marks
- Each paper is ramped so that there is an increase in demand through the paper
- Students are given Python source code files and any required input files
- Students are given the Programming Language Subset booklet for the exam
- Responses consist of amended source code files.





# Command word taxonomy

- There are only two command words used in this paper
- They are set out in the specification document on page 27.

Command word	Mark Tariff	Definition
Amend	5 to 15 marks	Requires changes or additions to code, or deletions or rearrangement of code.
Write	5 to 15 marks	Usually requires creation/manipulation of a program using a high-level programming language.

# Structure – Mark scheme

- Points-based
- Functionality Levels-based
  - Used in at least three questions
- Design Levels-based
  - Used in at least two questions
- Good Programming Practice Levels-based
  - Used in at least one question

## Functionality (levels-based mark scheme)

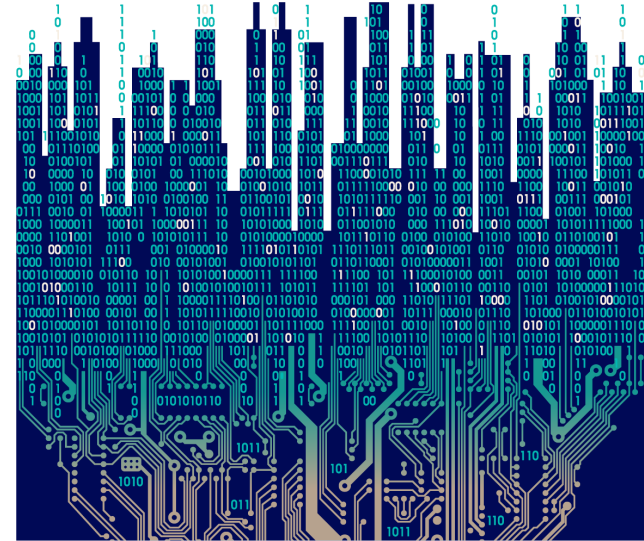
0	1	2	3	Max.
0	<b>Functionality (when the code is run)</b> <ul style="list-style-type: none"><li>• The component parts of the program are incorrect or incomplete, providing a program of</li></ul>	<b>Functionality (when the code is run)</b> <ul style="list-style-type: none"><li>• The component parts of the program are complete, providing a functional program that meets</li></ul>	<b>Functionality (when the code is run)</b> <ul style="list-style-type: none"><li>• The component parts of the program are complete, providing a functional program that fully meets</li></ul>	3

# Question types

- Name the part
  - Looking at code and answering questions about it
- Fix the errors
  - Correcting syntax, runtime, and logic errors
- Select the lines
  - Choosing the correct line of code from options
- Translate a flowchart
  - Implement the logic shown in the flowchart
- Complete the code
  - Making additions to existing lines or adding new lines
- Comment-first coding
  - Implementing existing logic, expressed as comments
- Parson's problem
  - Ordering given lines of code
- Open
  - Design and implement a solution to a problem

# Section 3 – Paper 2:

## Name the part



# Marked example

2023 Q01

**Suggested time: 10 minutes**

- 1 A program is being developed to show the average daily temperature and add up the costs of buying ice cream.
- It displays each temperature stored in an array of temperatures.
  - It adds up all the ice cream costs entered by the user, until the user enters 0.
  - It then calculates a discount. When the total cost is over 100.00, the discount is 10%. Otherwise, the discount is 5%.

Open file **Q01.py**

Amend the lines at the bottom of the code to give the:

- name of a constant used in the program
- name of an array used in the program
- line number of an initialisation of a variable with a real number
- line numbers for a selection construct
- line numbers for a repetition construct
- line numbers for an iteration construct
- line number for an instruction that outputs information to the screen.

Do **not** add any additional functionality.

Save your amended code file as **Q01FINISHED.py**

(Total for Question 1)

Question number	MP	Appx. Line	Answer
1			Award marks as shown.
	1.1	37	DISCOUNT_5 / DISCOUNT_10 (1)
	1.2	38	theTemperatures (1)
	1.3	40	10 / 11 / 12 (1)
	1.4	41	27 / 27, 29 / 27-30 (1)
	1.5	42	22 / 22-24 (1)
	1.6	43	17 / 17-18 (1)
	1.7	44	18 / 21 / 24 / 32 (1)

```

33 # -----
34 # =====> Write your answers here in the right-hand column
35 # Left                                     # Right
36 # -----
37 # Name of a constant used in the program      #DISCOUNT_5
38 # Name of an array used in the program        #theTemperatures
39 # Line number of an initialisation of a
40 # variable with a real number                  #21
41 # Line numbers for a selection construct       #27,29
42 # Line number(s) for a repetition construct   #22
43 # Line number(s) for an iteration construct   #17
44 # Line number for an instruction that outputs
45 # information to the screen                    #32,18,21,24

```

**6 marks**

# In the classroom

- Terms from the specification could be explored using flip cards
  - Give a code stimulus
  - A matching set of 10+ cards with descriptions
- Constructs and patterns from the PLS could be explored using cards
  - Card 1: a programming construct or pattern
  - Card 2: a matching name or description
- Create practice questions and mark schemes
  - Give students a code stimulus
  - Students write the question and mark scheme
  - Students exchange questions, answer questions, then mark response
  - Students discuss as required.



© Trevor Clifford / Pearson Education Ltd

# In the exam

- Read the question carefully to identify what is required (name, line number, etc.)
- Do not change the existing code, as this can make the line numbers change
- Use the PLS to look up terms, if required.

**Pearson Edexcel Level 1/Level 2 GCSE (9–1)**

Paper reference

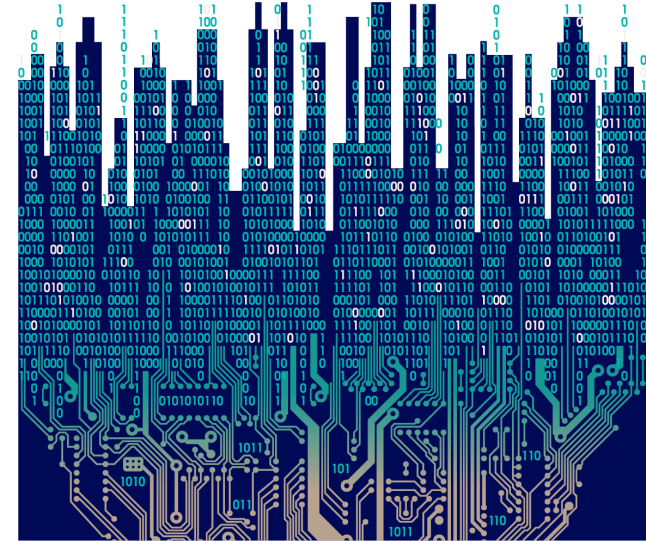
**1CP2/02**

**Computer Science**  
**PAPER 2: Application of Computational Thinking**  
**Programming Language Subset**  
**Version 4**

**PLS Booklet**  
You do not need any other materials.

# Section 3 – Paper 2:

## Fix the errors





# Marked example

2022 Q02

**Suggested time: 15 minutes**

- 2 A program uses turtle graphics to draw a simple image. This is the image that must be produced.

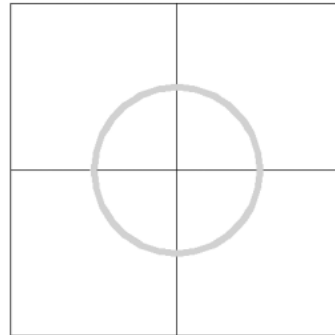
Both the circle and the outside square are centred on the horizontal and vertical grid lines. The outside square is  $400 \times 400$ . The circle is 200 across. The circle outline is coloured gold. All other lines are black.

The program has errors and does not work correctly.

Open file **Q02.py**

Amend the code to:

- add a comment to identify the data type of the argument to the `turtle.mode ()` subprogram call on original line 19  
`turtle.mode ("standard")`
- fix the `NameError` on original line 23  
`screen.setup (WIDTH, HEIGHT)`
- fix the `AttributeError` on original line 28  
`theTurtle = turtle.turtle ()`



Question number	MP	Appx. Line	Answer
2			Award marks as shown.
	2.1	19	Any comment with the word "string" and call (1)
	2.2	23	Name error – correct spelling of constant (1)
	2.3	28	Attribute error – Requires a capital letter (1)
	2.4	36	Type error – Remove argument to <code>&lt;turtle&gt;.pendown ()</code> (1)
	2.5	42	Logic error – Move vertical grid line back to origin (1) <code>theTurtle.setpos (0, 200)</code>
	2.6	48	Logic error – Correct length of vertical grid line (1) <code>theTurtle.forward (400)</code>
	2.7	56	Logic error – Correct heading for starting point of square (1) <code>theTurtle.setheading (90)</code>
	2.8	68	Control pen size with a constant (1) <code>theTurtle.pensize (BIG)</code>
	2.9	71	Set the pen colour to "gold" (1) <code>theTurtle.pencolor ("gold")</code>
	2.10	78	Hide the turtle (1) <code>theTurtle.hideturtle ()</code>

```
13 # -----
14 # Main program
15 # -----
16 # Setup the turtle environment
17 # =====> Add a comment to identify the data type of the argument to the turtle.mode () subprogram call
18 #
19 turtle.mode ("standard") #
20 screen = turtle.Screen ()
21
22 # =====> Fix the NameError
23 screen.setup (WIDTH, HEIGHT)
24 turtle.screensize (WIDTH, HEIGHT)
25
26 # Prepare the turtle
27 # =====> Fix the AttributeError
28 theTurtle = turtle.turtle ()
29 theTurtle.penup ()
```

8 marks

# In the classroom

- Create similar questions and mark schemes
  - Students to work in pairs using their IDE
  - Discuss what is causing the error
  - Propose a fix
  - Test the fix by executing the code
- Explore the order in which errors should be fixed
  - Syntax, runtime, logical
- Set stopwatches to see how quickly all the errors can be found and fixed.

## Specimen 3 Q02

### Suggested time: 15 minutes

- 2 A programmer has started to write a program, but it does not work correctly.

The program should ask users to enter their initials.

The program should report an error when the user enters fewer than two initials or more than three initials or when the characters are not alphabetic.

When the input is valid, the initials should be reported back to the user in uppercase.

The program should allow the user to go again as long as the user enters a 'Y' or 'y' when asked.

Open file **Q02.py**

Amend the code to:

- fix the syntax error on original line 5  
`initials =`
- fix the syntax error on original line 17  
`else if (len (initials) > 3):`
- fix the syntax error on original line 19  
`else`
- change the Boolean operator to fix the logic error on original line 11  
`while ((a == "Y") and (a == "y")):`
- add a Boolean operator to fix the logic error on original line 13  
`if (initials.isalpha ()):`
- change the relational operator to fix the logic error on original line 15  
`elif (len (initials) <= 2):`
- amend the assignment to fix the logic error on original line 23  
`a == input ("Would you like to go again? ")`

# In the exam

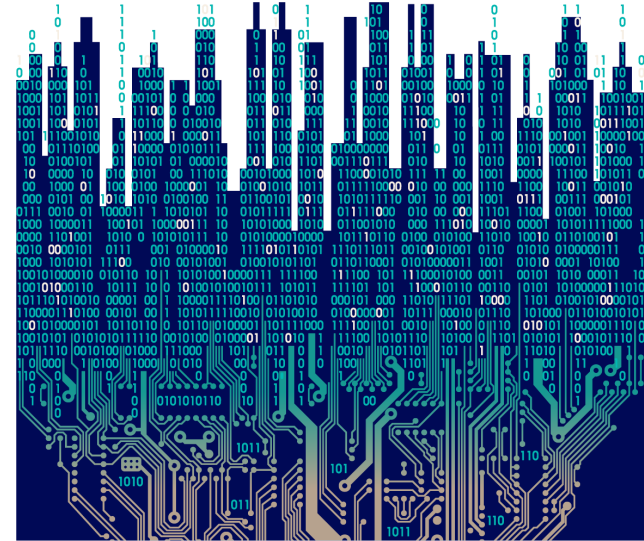
- Read the question carefully to identify what is required
- Identify an order to tackle the errors
- Change only the lines indicated in the question paper
- Change one item at a time
- Execute the code often to check on progress.



© rastudio / 123RF

# Section 3 – Paper 2:

## Select the lines



# Marked example

Suggested time: 20 minutes

2023 Q03

3 A program is used in a shop that sells building materials.

The program reads in data about screws from a file. The data file is provided.

The program counts the number of copper screws.

The program stores the names of 12 bricks in an array. It writes the names of the bricks to a different file, one name per line. Brick names must be in uppercase.

The program displays this output on the screen.

Total screws: 26 Copper screws: 5  
Wrote 12 brick names to file

The output shows 26 screws were read from the file, and five are made from copper. It also shows 12 brick names were written to the file.

Open file **Q03.py**

Amend the code to

You will need to:

- amend some lines of code
- choose between different options
- run the program and check the results

Question number	MP	Appx. Line	Answer	Additional guidance
3			Award marks as shown.	Only one response for MCQ. Do not add more than one line uncommented.
			Constants	
	3.1	7	Add "Screws.txt", including quotes INPUT_FILE = "Screws.txt" (1)	• Must be name supplied file, with "Screws.txt"
	3.2	10	Add .txt to Bricks file name OUTPUT_FILE = "Bricks.txt" (1)	• Allow .txt after quotes • Allow .csv
			Global variables	
	3.3	16	Add brickTable as name of array before assignment symbol brickTable = ["Rustic", ...] (1)	
	3.4	31	Choose integer initialisation total = 0 (1)	
	3.5	36	Choose string initialisation: outline = "" (1)	
			Processing copper screws	
	3.6	50	Choose constant file name and open for read only: inFile = open (INPUT_FILE, "r") (1)	
	3.7	56	Choose result of find() != -1 if (line.find (SPECIFIED_MATERIAL) != -1): (1)	
	3.8	61	Add code to increment total by one total + 1 (1)	• Allow total += 1
	3.9	64	Choose closing that matches the correct opening on line 48	

```
28 # =====> Choose the correct value to initialise the variable
29 #total = 0.0
30 #total = ""
31 total = 0
32 #total = True
33
34 # =====> Choose the correct value to initialise the variable
35 outline = False
36 #outline = ""
37 #outline = 0.0
38 #outline = 0
39
40 # -----
41 # Main program
42 # -----
43
44 # Process the screws
45
46 # =====> Choose the correct line to open the file
47 #inFile = open ("Screws", "r")
48 #inFile = open ("Screws", "a")
49 #inFile = open (INPUT_FILE, "a")
50 inFile = open (INPUT_FILE, "r")
```

13 marks





# In the classroom

- Create similar questions and mark schemes
  - Students to work in pairs using their IDE
  - Discuss which option of the four is the best
  - Uncomment the selected line
  - Test the fix by executing the code
- Compare the code outputs to those required in the question paper.

## Specimen 2 Q02

Suggested time: 15 minutes

- 2 A program is needed to simulate the pedestrian crossing lights that tell pedestrians when to stop and when to walk. The program repeats this sequence three times.

The hand signal is displayed for 15 seconds.	
The walk signal is displayed for 10 seconds.	
The walk signal, along with a countdown, is displayed for 20 seconds.	 

(Source: © Pearson Asset Library)

Open file **Q02.py**

Amend the code to make the program work and produce the correct output.

You will need to choose between alternative lines of code given on original lines:

- 5 and 6
- 29 and 30
- 33 and 34
- 37 and 38
- 40 and 41
- 44 and 45
- 47 and 48

Make a choice by removing the # at the beginning of the line you choose to execute.

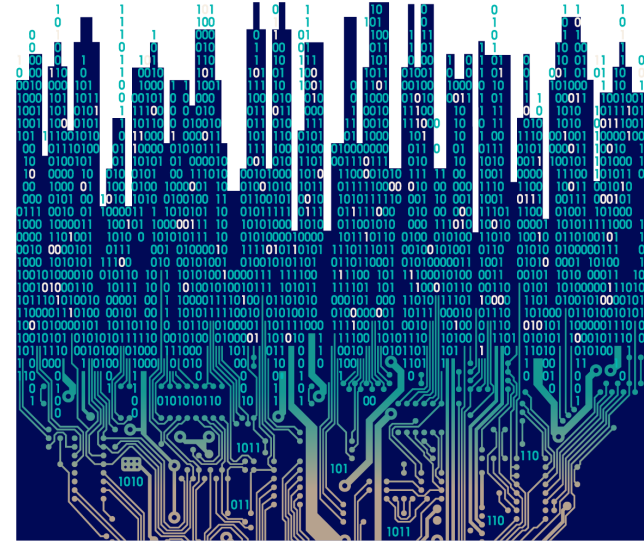
# In the exam

- Read the question carefully to identify what is required
- Read all the comments in the code file to understand the objective of the program
- Change only one line in each group, by removing the comment symbol (#)
- Execute the code often to check on progress
- Check the outputs to make sure they match the requirements in the question paper.



# Section 3 – Paper 2:

## Translate a flowchart





# Marked example

Specimen 1 Q03

Suggested time: 15 minutes

- 3 The flowchart on the facing page is for an algorithm that performs a presence check validation and length check validations on a name entered by the user. No other validation is required.

The program has these requirements:

- add a comment to identify the code for
- add a comment to identify the code for
- test the functionality of the program us

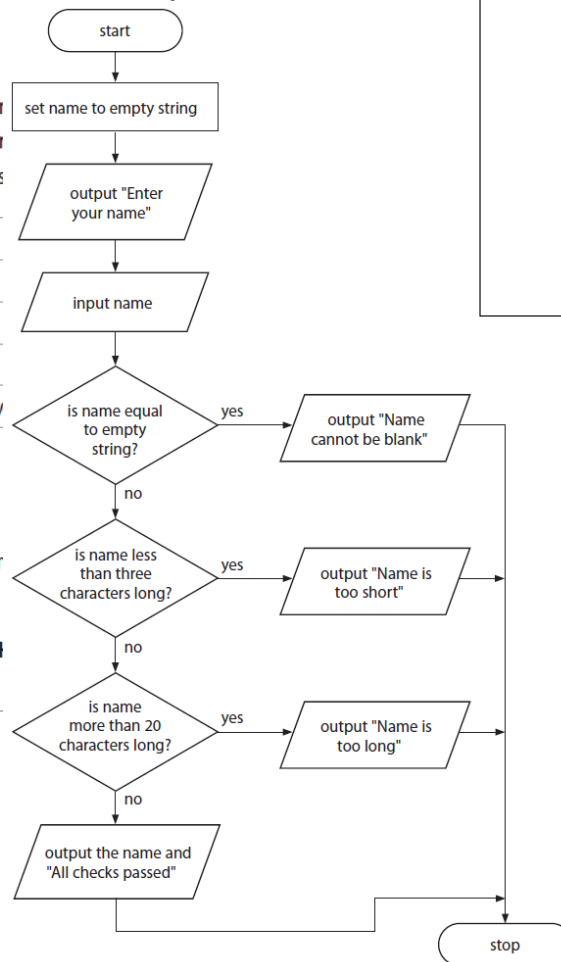
Input
Empty string
Ab
Rebecca
abcdefghijklmnopqrstuvwxyz

Open file Q03.py

Write the code to implement the algorithm

Do **not** add any additional functionality.

Save your amended code file as Q03FINISH



Question number	Answer
3	<p>Award marks as shown.</p> <ul style="list-style-type: none"><li>• Variable 'name' set to "" (1)</li><li>• Get value of 'name' from user using 'input' (1)</li><li>• At least one 'if' statement used (1)</li><li>• Presence check using "" / len (name) == 0 (1)</li><li>• Length check using '&lt; 3' (1)</li><li>• Length check using '&gt; 20' (1)</li><li>• Use of 'if...elif...elif...else' rather than separate 'if' (1)</li><li>• Presence check identified in comments and at least one length check identified in comments (1)</li><li>• Both length checks identified in comments (1)</li><li>• Functionality tested (1)</li></ul>

```
1 # -----
2 # Global variables
3 # -----
4 name = ""
5
6 # -----
7 # Main program
8 # -----
9 name = input ("Enter your name ")
10 if name == "":
11     print ("Name cannot be blank")
12 elif name < 3:
13     print ("Name is too short")
14 else:
15     if len (name) > 20:
16         print ("Name is too long")
17     else:
18         print ("All checks passed")
19
```

5 marks

# In the classroom

## Specimen 1 Q04

- Create flowcharts from existing code samples
- Arrange flowcharts to create a functional solution
  - Give only required steps
  - Give extra steps
- Create similar questions and mark schemes.

**Suggested time: 20 minutes**

- 4 The flowchart on the facing page is for an algorithm that identifies the minimum and the maximum numbers entered by the user. The numbers are positive real numbers less than 100 000. It also calculates the mean of all the numbers. The results are displayed for the user.

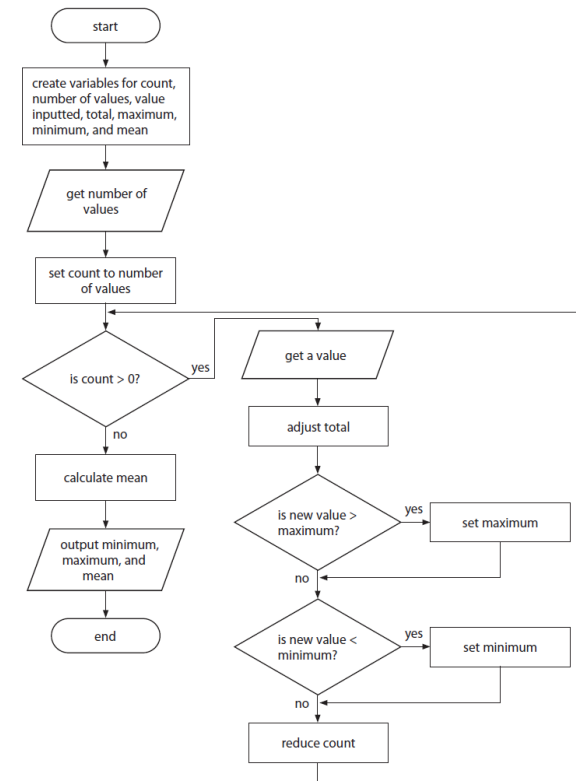
Open file **Q04.py**

Write the code to implement the algorithm in the flowchart.

Do **not** add any additional functionality.

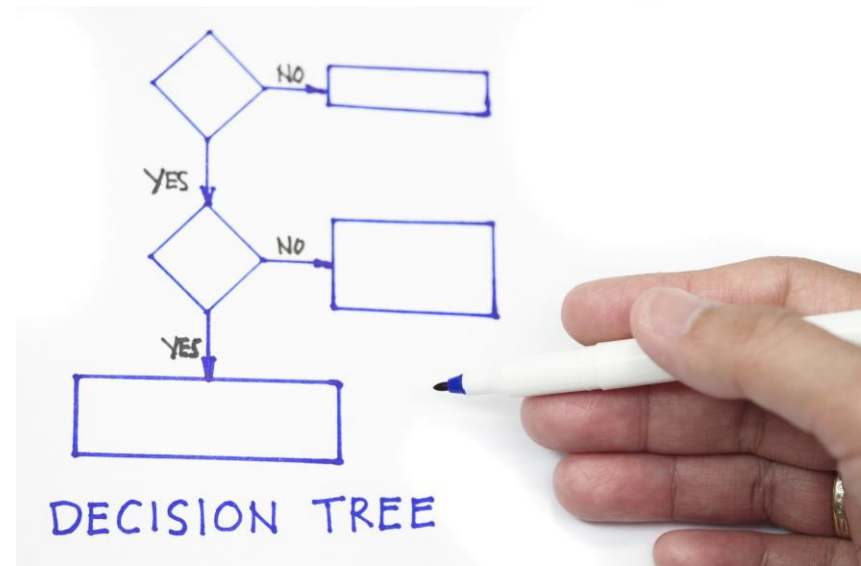
Save your amended code as **Q04FINISHED.py**

(Total for Question 4 = 15 marks)



# In the exam

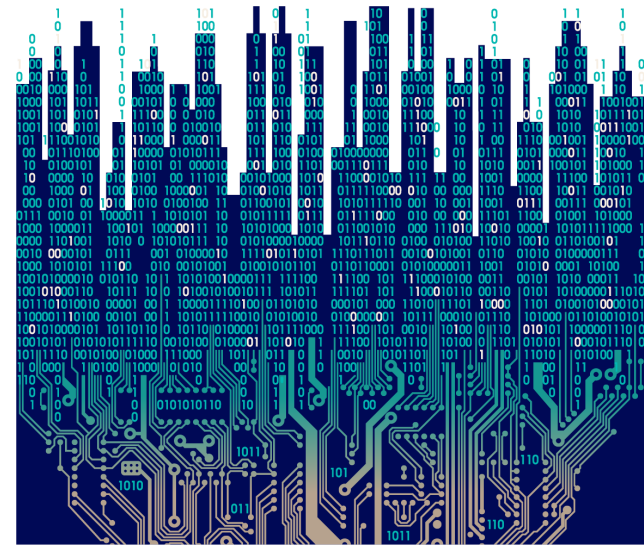
- Read the question carefully to identify what is required
- Work with a small part of the flowchart at a time
  - Create the code for a small part, such as input and outputs or a decision
  - Check that small part translates and executes
  - Repeat with another small part
- Add a comment or two to explain the logic
- Check the outputs to make sure they match the requirements in the question paper.



© RAGMA IMAGES / Shutterstock

# Section 3 – Paper 2:

## Complete the code



# Marked example

Specimen 3 Q01

- 1 A program must calculate the circumference of a circle. The user enters the radius of the circle. A radius of zero or less is invalid.

The formula to calculate the circumference of a circle is:

$$\text{circumference} = 2\pi r$$

- $\pi$  is the constant Pi
- $r$  is the radius.

Open file **Q01.py**

Amend the code to add or complete lines to:

- import the math library
- create two variables
- take input from the user
- check for an invalid input of zero or less
- display a message to tell the user the input is invalid
- calculate the circumference
- round the circumference to three decimal places using `round()`

Do **not** add any additional functionality.

Save your amended code file as **Q01FINISHED.py**

(Total: 9 marks)

Question number	Answer
1	<p>Award marks as shown.</p> <ul style="list-style-type: none"><li>• Add a line to import the math library Original: &lt;Blank&gt; Amended: <code>import math</code></li><li>• Create and set an integer variable <code>radius</code> Original: &lt;Blank&gt; Amended: <code>radius = 0</code></li><li>• Create and set a real variable <code>circumference</code> (1) Original: &lt;Blank&gt; Amended: <code>circumference = 0.0</code></li><li>• Complete the line to take input from the user Original: <code>radius =</code> Amended: <code>radius = int(input("Please enter the radius of the circle: "))</code> Call to <code>input()</code>, with a prompt ("Please enter the radius of the circle: ") Call to <code>int()</code> to convert string to integer</li><li>• Complete the line to validate for invalid input Original: <code>radius</code> Amended: <code>radius &lt;= 0</code></li><li>• Add a line to print an invalid input message (1) Original: &lt;Blank&gt; Amended: <code>print("Invalid radius")</code></li><li>• Complete the calculation of the circumference (1) Original: <code>circumference =</code> Amended: <code>circumference = 2 * math.pi * r</code></li><li>• Complete the line to round circumference to three decimal places (1) Original: <code>circumference =</code> Amended: <code>circumference = round(circumference, 3)</code> Call to <code>round()</code> (1) Correct parameters to <code>round()</code> (1)</li></ul>

```
9 # -----
10 # =====> Create an integer variable named radius and set it to 0
11 radius = 0
12
13 # =====> Create a real variable named circumference and set it to 0.0
14 circumference = 0
15
16 # -----
17 # Main program
18 # -----
19 # =====> Complete the line to assign an integer, input by
20 # the user, to the variable radius
21 radius = int(input("Please enter the radius of the circle: "))
22
23 # =====> Complete the code in the brackets to check for
24 # an invalid radius of zero or less input by the user
25 if (radius <= 0):
26     # =====> Add a line to tell the user the entry is invalid
27     print("Invalid")
28 else:
29     # =====> Complete the calculation of the circumference
30     circumference = 2 * (math.pi * radius)
31
32 # =====> Complete the line to round circumference to three
33 # decimal places using the round() function
34 circumference = round(circumference, 3)
35
36 print ("The circumference is", circumference)
```

9 marks

# In the classroom

2022 Q01

- Create similar questions and mark schemes
  - Start with a functional solution and remove parts
- Investigate what happens when an inaccurate change is provided
- Students work in pairs to discuss what the missing code might be.

Suggested time: 15 minutes

- 1 A program is required to convert numbers entered by the user to their alphabetic equivalent. Only numbers from 5 to 30 are valid.

Adding 60 to the number and then applying the function `chr()` generates the equivalent ASCII code for an uppercase letter.

The table shows accurate test data for a functional program.

Input	Output
4	Invalid input
5	5 is equal to A
22	22 is equal to R
30	30 is equal to Z
31	Invalid input

Open file **Q01.py**

Amend the code to:

- create an integer variable named `num` and set it to 0
- take the input from the user and convert it to an integer
- check that the inputted number is between 5 and 30
- add 60 to the variable `num` and assign the result to the variable `decimalCode`
- join strings together with concatenation
- display an error message.

Do **not** add any additional functionality.

Save your amended code file as **Q01FINISHED.py**

(Total for Question 1 = 10 marks)

# In the exam

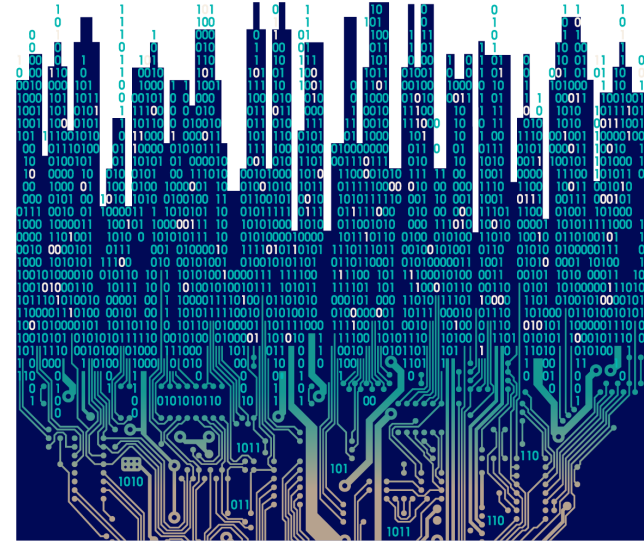
- Read the question carefully to identify what is required
- Change only the lines indicated in the question paper
- Change one item at a time
- Execute the code often to check on progress
- Use the debugger to stop code near the corrected line to check if it is working correctly.



© mohdizuan / Shutterstock

# Section 3 – Paper 2:

## Comment-first coding





# Marked example

Specimen 1 Q05

Suggested time: 25 minutes

- 5 A program is needed to track the number and health scores of native British trees. The tree data is stored as structured data. These are implemented as three lists.

The program needs to be amended to meet these requirements:

- construct a record for each type of tree consisting of name, number and health score
- use commas to separate fields
- write each record to a separate line in an output file.

This is what your final output file should look like. T output file.

```
1 Maple,1357,0.15
2 Ash,8421,0.18
3 Sycamore,9287,0.73
4 Birch,1043,0.38
5 Hazel,3743,0.92
6 Willow,2948,0.24
7 Oak,10826,0.78
```

Open file **Q05.py**

Amend the code to:

- process the lists to generate the output file.

Do **not** add any additional functionality.

Use comments, white space and layout to make the program easier to read and understand.

Save your amended code file as **Q05FINISHED.py**

Question number	Answer
5	<p>Award marks as shown.</p> <ul style="list-style-type: none"><li>Use of 'for' or 'while' loop to process every tree</li><li>Use of string concatenation to build output string</li><li>Output string contains 3 items in order (name, number and score) (1)</li><li>Output string fields separated by commas (1)</li><li>Line feed added to end of output string (1)</li><li>Use of same or equivalent index variable for tree name, number and score. (1)</li><li>Use of white space and comments aids readability (1)</li></ul> <p>Levels-based mark scheme to a maximum of 6, for</p> <ul style="list-style-type: none"><li>Solution design (3)</li><li>Functionality (3)</li></ul>

```
1 # -----
2 # Constants
3 # -----
4 OUTFILE = "Trees.txt"
5
6 # -----
7 # Global variables
8 # -----
9 treeNames = ["Maple", "Ash", "Sycamore", "Birch", "Hazel", "Willow", "Oak"]
10 treeCounts = [1357, 8421, 9287, 1043, 3743, 2948, 10826]
11 treeScore = [0.15, 0.18, 0.73, 0.38, 0.92, 0.24, 0.78]
12 # ==> Write your code here
13 output = ""
14 i = 0
15 # -----
16 # Main program
17 # -----
18 # ==> Open the output file for writing
19 file = open(OUTFILE, "w")
20
21 # ==> Use iteration to process each item in treeNames
22 for i in range(len(treeNames)):
23     # ==> Create a comma separated value output line
24     output = (treeNames[i] + ", " + str(treeCounts[i]) + ", " +
25              str(treeScore[i]) + "\n")
26     # ==> Write the line to the file
27     file.writelines(output)
28 # ==> Close the output file
29 file.close()
```

12 marks

(Total for Question 5 = 13 marks)

# In the classroom

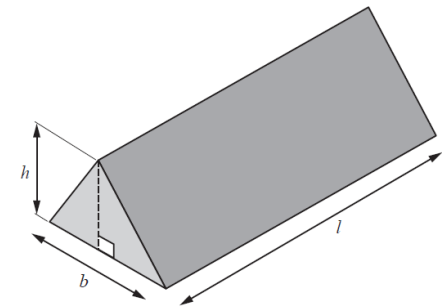
2023 Q04

- Create similar questions and mark schemes
  - Start with a functional solution that includes comments
  - Remove some or all of the code
- Students work in pairs to discuss the logic and structure of the solution shown in the comments
  - Why is the functionality in each comment needed?
  - Would it work if the logic were rearranged?

**Suggested time: 25 minutes**

- 4 A program is required to calculate the volume of a prism. All dimensions are entered by the user. The dimensions are decimal numbers greater than 0

The volume of this prism is the area of the triangle multiplied by the length of the prism.



The formula to calculate the area of the triangle is:

$$A = \frac{1}{2} \times b \times h$$

- $A$  is the area of the triangle
- $b$  is the width of the base of the triangle
- $h$  is the height of the triangle

The formula to calculate the volume of this prism is:

$$V = A \times l$$

- $V$  is the volume of the prism
- $A$  is the area of the triangle
- $l$  is the length of the prism

# In the exam

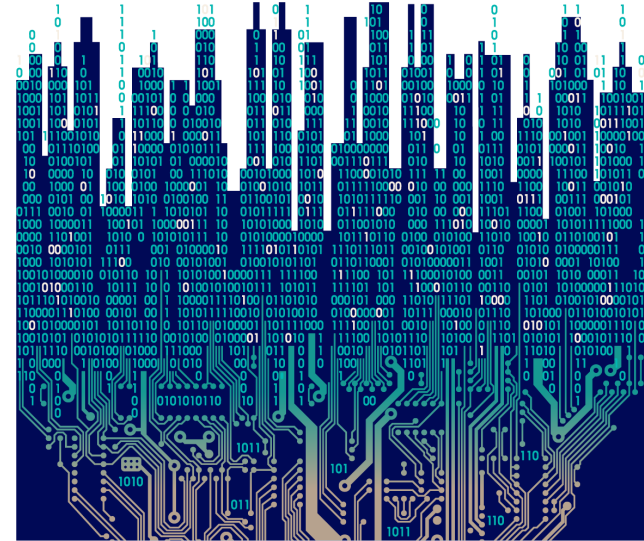
- Read the question carefully to identify what is required
- Read the comments to understand the overall logic before adding code
- Be sure to keep the indentation, if provided, as it helps with the logic
- Develop one section of the code at a time
- Execute the code often to check on progress
- Use the debugger to make sure each section of code is correct.



© Spectral-Design / Shutterstock

# Section 3 – Paper 2:

## Parson's problem



# Marked example

Specimen 1 Q03

Suggested time: 20 minutes

- 3 A program calculates the bus fare for passengers. The type of fare is entered from a menu. The fare is calculated and displayed. There is a choice to exit the program instead of calculating a fare.

The program is being developed. Some of the lines of code in the program are missing. Some sections of the program include different code options and only one option should be included in the program.

Open file Q03.py

Amend the code to make the program work and produce the correct output.

You will need to rearrange lines of code or choose between alternative lines of code.


Do not change the functionality of the given lines of code.

Do not

Save your

```
80 # -----
81 # Main program
82 # -----
83
84 # ==> Here is a group of jumbled lines
85 # ==> Order the lines by moving them into their correct positions
86 # ==> Indentation Levels are shown for you with comments
87 # ==> The number of lines to place is shown in brackets
88
89 # if (choice != "Q"):
90 # while (choice != "Q"):
91 #     print (layout.format (fare))
92 #     choice = getUserChoice ()
93 # print ("Goodbye")
94 #     fare = calcFare (choice)
95 #     showMenu ()
96
```

```
37 def getUserChoice ():
38     userChoice = ""
39     validChoice = False
40
41     # ==> Choose one line
42     while (validChoice):
43         #while (not validChoice):
44             userChoice = input ("Please choose a fare: ")
45             userChoice = userChoice.upper ()
46             # ==> Choose one long statement of 3 lines
47             if ((userChoice != "Q") and (userChoice != "A") and
48                 (userChoice != "B") and (userChoice != "C") and
49                 (userChoice != "D") and (userChoice != "E")):
50                 #if ((userChoice == "Q") or (userChoice == "A") or
51                     #     (userChoice == "B") or (userChoice == "C") or
52                     #     (userChoice == "D") or (userChoice == "E")):
53                     print ("Invalid entry")
54             else:
55                 # Choose one line
56                 #validChoice = False
57                 validChoice = True
58
59     return (userChoice)
```



14 marks

# In the classroom

SAM Q04

- Create similar questions and mark schemes
  - Start with a functional solution
  - Rearrange the lines of code
- Students work in pairs to discuss the logic required in the solution
  - Does the indentation help identify iteration, repetition, or selection?
  - Does the initialisation of variables stand out?
- Good IT skills
  - Make a copy of the original code
  - Open the copy and amended side by side
  - Use copy/paste effectively
  - Check indentation after moving.

Suggested time: 20 minutes

- 4 A program takes a year group as input and outputs the stage of education the year group belongs to.

The program loops continually until the user inputs 0 to stop the program.

Input	Output
0	Exits program
1, 2, 3, 4, 5, 6	Primary
7, 8, 9, 10, 11	Secondary
12, 13	College

The lines of code in the program are mixed up.

Open file **Q04**.

Amend the code to make the program work and produce the correct output. You will need to rearrange the lines.

Use comments, white space, indentation and layout to make the program easier to read and understand.

Do **not** change the functionality of the given lines of code.

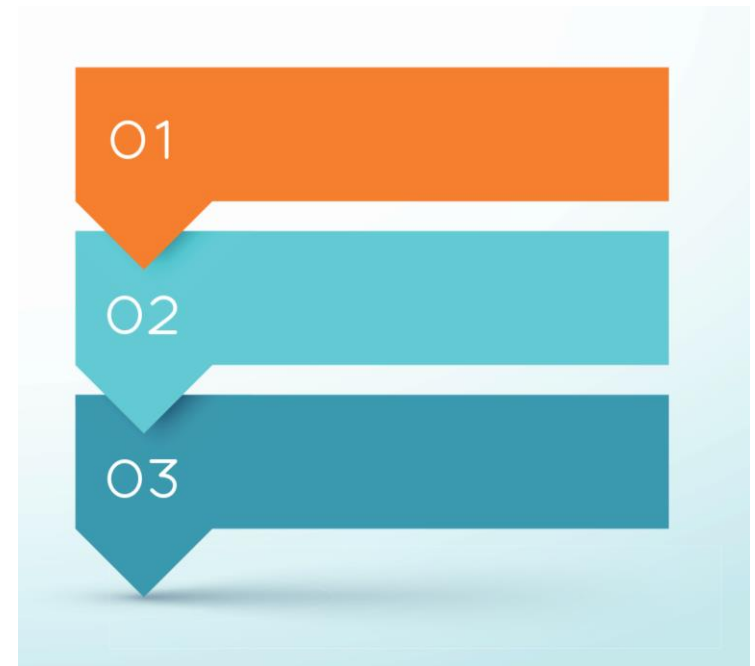
Do **not** add any additional functionality.

Save your amended code file as **Q04FINISHED.py**

(Total for Question 4 = 15 marks)

# In the exam

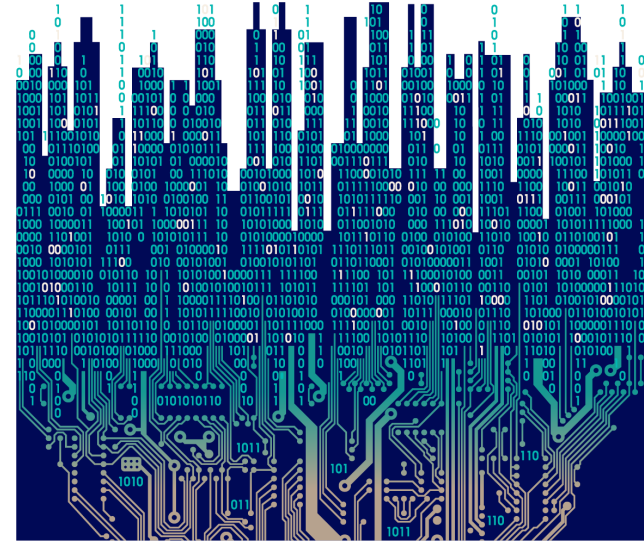
- Read the question carefully to identify what is required
- Read the comments to understand the overall logic before rearranging the code
- Be sure to keep the indentation, as it helps with the logic
- Isolate the variable initialisations
- Develop one section of the code at a time
- Execute the code often to check on progress
- Use the debugger to stop execution to prevent going into the section of code not being worked with at any time
- Remove all the syntax errors.



© olliethedesigner / 123RF

# Section 3 – Paper 2:

## Open





# Marked example

Specimen 2 Q06

Suggested time: 30 minutes

- 6 A program controls a sorting machine for packages based on weight. Gates are opened to direct packages down different paths.

Weight in grams	Path
1 to 100	Green
101 to 750	Yellow
751 to 1000	Red

The maximum number of packages that can be processed in any one hour is 100. At the end of each hour, the program reports sorting information.

Open file **Q06.py**

Write a program to meet the following requirements:

## Inputs

- Prompt for and accept the number of packages, 1 to 100 inclusive.
  - When invalid input is received, the number of packages should be set to 100

## Process

- Create a data structure for each coloured path.
- Fill each data structure with random weights, between 1 and 1000, inclusive. The random weights are to be generated by the program.
- Find the number of weights in each data structure.
- Find the maximum weight in each data structure.

## Outputs

- Display an output line for each path with the colour, the number of weights, and the maximum weight.
- Outputs should be formatted as English sentences with appropriate spacing between words and punctuation.

Decompose the solution, using one or more subprograms.

Use comments, white space and layout to make the program easier to read and understand.

Do **not** add any additional functionality.

Save your amended code as **Q06FINISHED.py**

Question number	Answer	Additional guidance	Mark
6	<p>Award marks as shown.</p> <p>Points-based mark scheme:</p> <ul style="list-style-type: none"><li>Accepts string input and converts to integer (1)</li><li>One-dimensional data structures (list) created for each of the colour paths (1)</li><li>Use of concatenation/&lt;string&gt;.format() to form English sentences with spacing and punctuation (1)</li><li>One or more subprograms created and called (1)</li><li>Import random and use of random.randint() (1)</li><li>Range check used for validation with invalid items defaulting to 100 (1)</li></ul> <p>Levels-based mark scheme to a maximum of 9, from:</p> <ul style="list-style-type: none"><li>Solution design (3)</li><li>Good programming practices (3)</li><li>Functionality (3)</li></ul>	<p>Considerations for levels-based mark scheme:</p> <ul style="list-style-type: none"><li>[6.1.2] Write in a high-level language</li><li>[6.2.2] Main program code is laid out in clear sections; white space is used to show different parts of the solution/functionality; variable names are meaningful; comments are provided and are helpful</li><li>[6.2.2] Use of iteration to find maximum weight</li><li>[6.4.1] Messages match content of data structures</li><li>[6.1.6] Functions correctly for all numeric input (i.e. edge conditions and defaulting to 100)</li><li>[6.1.1] Use decomposition to solve problem and create solution</li><li>[6.2.2] Use of 'for' loop to iterate over a data structure, rather than a 'while' loop</li><li>[6.3.1] Conversion of input types to those required by program, e.g. two strings and two integers</li><li>[6.1.6] Tracking of maximum value for</li></ul>	(15)

```
35         greenPath.append (weight)
36     elif weight <= 750:
37         yellowPath.append (weight)
38     else:
39         redPath.append (weight)
40
41     # Do the outputs
42     print ("Green: ", "Num = ", len (greenPath), "Max = ", max (greenPath))
43     print ("Yellow: ", "Num = ", len (yellowPath), "Max = ", max (yellowPath))
44     print ("Red: ", "Num = ", len (redPath), "Max = ", max (redPath))
45
```

13 marks

(Total for Question 6 = 15 marks)

# In the classroom

Specimen 3 Q06

- Create similar questions and mark schemes
  - Start with a functional solution
  - Remove the code
  - Remove any variables, if they need to be created in the code
- Students work in pairs to discuss the logic required in the solution
  - What do the variables and data structures look like?
  - Where is a loop used?
  - Where is selection used?
  - Can a subprogram be used to help the abstract away logic?

**Suggested time: 25 minutes**

- 6** A program is needed to produce codes for labels that identify the artist of works in an exhibition.

Records for each artist are stored in a two-dimensional data structure, implemented as a list. The fields for each record are first name, last name, and year of birth.

The code for each artist is constructed by joining the first letter of the last name, the first letter of the first name, and the year of birth.

For example, a coded label for ("Andy", "Warhol", 1928) is 'WA1928'.

Open file **Q06.py**

Write a program to meet the following requirements:

- create a code for each artist in 'theArtists'
- store all codes in the data structure named 'theLabels'
- display the labels for all the artists, one label per line
- find and display the name and year of birth of the youngest artist
- ensure the code works for any number of artists.

Do **not** add any additional functionality.

Use comments, white space and layout to make the program easier to read and understand.

Save your amended code as **Q06FINISHED.py**

---

(Total for Question 6 = 15 marks)

# In the exam

- Read the question carefully to identify what is required
- Identify the main parts of the solution
  - Subprograms, loops, data structures, and variables
- Develop one small section of the code at a time
- Use items provided in the code file
- Execute the code often to check on progress
- Remove all the syntax errors.



© Nenov Brothers Images / Shutterstock

# Take away

- Classroom
  - Use published materials as part of regular teaching
  - Run a mock, under timed conditions, using one of the published past papers
- Exam
  - Address the bullets in the question
  - Do not overcomplicate matters; the problems can be solved with simple constructs
  - Do not change the given code, unless instructed, even if there is a better, shorter, more efficient solution
  - Work on a small section of code at a time
  - Make sure the code doesn't crash, even if it doesn't work properly
  - Always add a comment or two; don't be excessive
  - Save finished code as a new name, so the original is not lost.

## Pearson Edexcel Level 1/Level 2 GCSE (9–1)

Time 2 hours

Paper  
reference

**1CP2/02**

### Computer Science

#### PAPER 2: Application of Computational Thinking

##### You must have:

- a computer workstation with appropriate programming language code editing software and tools, including an IDE that you are familiar with that shows line numbers
- a 'STUDENT CODING' folder containing code and data files
- printed and electronic copies of the Program Language Subset (PLS) document.

## **Question 4**

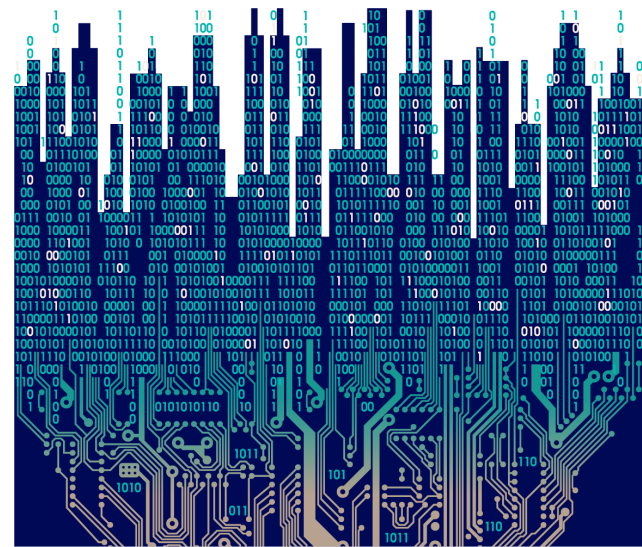
Let's see how we can help each other.

### **Chat window:**

1. What methods do you use to prepare for Paper 2?

# Section 4 – Logistics for Paper 2

[Pre-recorded session](#) on the support and guidance for delivering the assessment, key preparation activities and what to expect on exam day.



# Highlights – administrative

- No installation of any special software from Edexcel
- We can help you maximise capacity of candidates in any room
  - JCQ guidelines are quite old so judicious use of screen shaders and room partitions can help with capacity
- Staggered sittings are permitted using the exam clash processes that Exams Officers already know about.

Guidance documents can be found on this [link](#):

<https://qualifications.pearson.com/en/qualifications/edexcel-gcses/computer-science-2020.coursematerials.html#%2FfilterQuery=Pearson-UK:Category%2FForms-and-administration>

# Highlights – technical

- No internet, unless this is required to run your tools, in which case you need to contact us to explain how security is managed
- Each student needs an exam profile (user area) that is only accessible during the exam
- Student seed code files are made available on the morning of the PM exam
- Following the exam, you must zip, and send all candidate work using Learner Work Transfer.

Paper 2 Delivery Webinars – usually one before and one after the entry deadline

## Centre guidance



[1CP2 02 Digital Submission Guidance v1.2](#)

| PDF 387.5 KB | 24 April 2023



[1CP2 02 Instructions for the Conduct of the Examination \(ICE\) v1.6 - Summer 2023](#)

| PDF 1.0 MB | 05 April 2023



[1CP2 02 IT Technician Checklist](#)

| PDF 487.7 KB | 06 February 2023



[1CP2 02 Examination Officer Checklist](#)

| PDF 640.0 KB | 03 February 2023



[1CP2 02 Invigilator Checklist](#)

| PDF 465.9 KB | 10 January 2023



[Guidance for Chrome Schools](#)

| PDF 232.1 KB | 10 November 2022



# Other resources

- Specification
- Getting Started Guide
  - Expands on some of the specification points
- Programming Language Subset
  - The constructs in Python 3 that the examiners use to set the questions
- Good Programming Practice Guide
  - Expands on the content of the PLS, with in-depth information and examples
- Sample Assessment Materials (SAM)
- Specimen 1, 2, and 3
  - Include exemplar assessment pack, with marked responses
- Live Series (from 2022)
  - Question papers
  - Mark schemes
  - Student code
  - Principal examiner's report

Switching to Pearson:

<https://qualifications.pearson.com/en/qualifications/edexcel-gcses/computer-science-2020/switch-to-pearson.html>

# Computer Science – subject page

The Computer Science support page can be found [here](#).

You can now register for the support sessions for the 2406 series:

**Wednesday 13 March 2024, 4pm:**  
<https://events.teams.microsoft.com/event/e492b48e-10b4-4aae-9892-3d12aaf54067@8cc434d7-97d0-47d3-b5c5-14fe0e33e34b>

**Wednesday 24 April 2024, 4pm:**  
<https://events.teams.microsoft.com/event/719d0236-dd57-4820-8b07-f0c50e2c340b@8cc434d7-97d0-47d3-b5c5-14fe0e33e34b>

## Your subject advisor

### Tim Brady

Computer Science and ICT Subject Advisor



**Email :** [TeachingComputerScience@pearson.com](mailto:TeachingComputerScience@pearson.com)  
[TeachingICT@pearson.com](mailto:TeachingICT@pearson.com)

**Facebook :** BTEC Tech Award in DIT Facebook group  
BTEC Firsts in I&CT Facebook group  
T-Level Digital Facebook group  
GCSE Computer Science Facebook group

**Phone :** +44 (0) 344 463 2535  
(Teaching Services team | Mon-Fri, 9am-5pm GMT)

➤ [Book an appointment with your subject advisor](#)

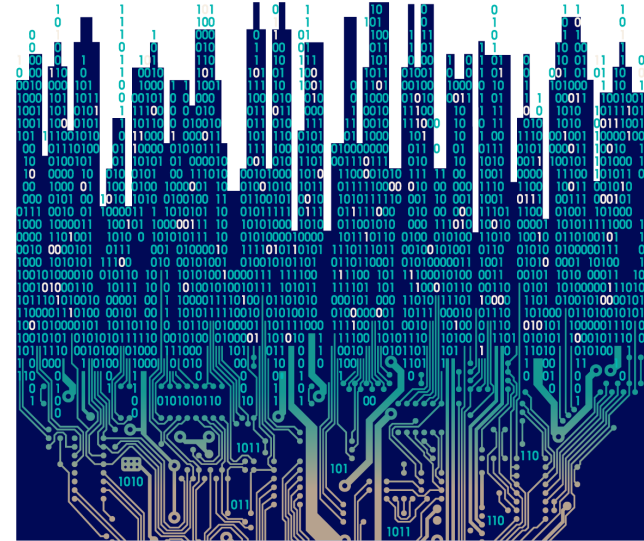


➤ [Access the customer support portal](#)

➤ [Access the Pearson community](#)

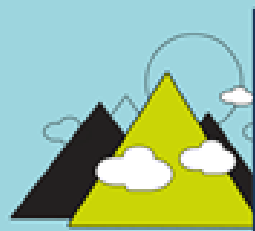
➤ [Sign up to receive subject advisor updates](#)

# Section 6 – Q&A



# Find out more

For more courses see our [Pearson Professional Development Academy](#).



Professional  
Development  
**Academy**

Transforming  
training for  
everyone.



Pearson