Pearson Edexcel

# GCSE (9-1)

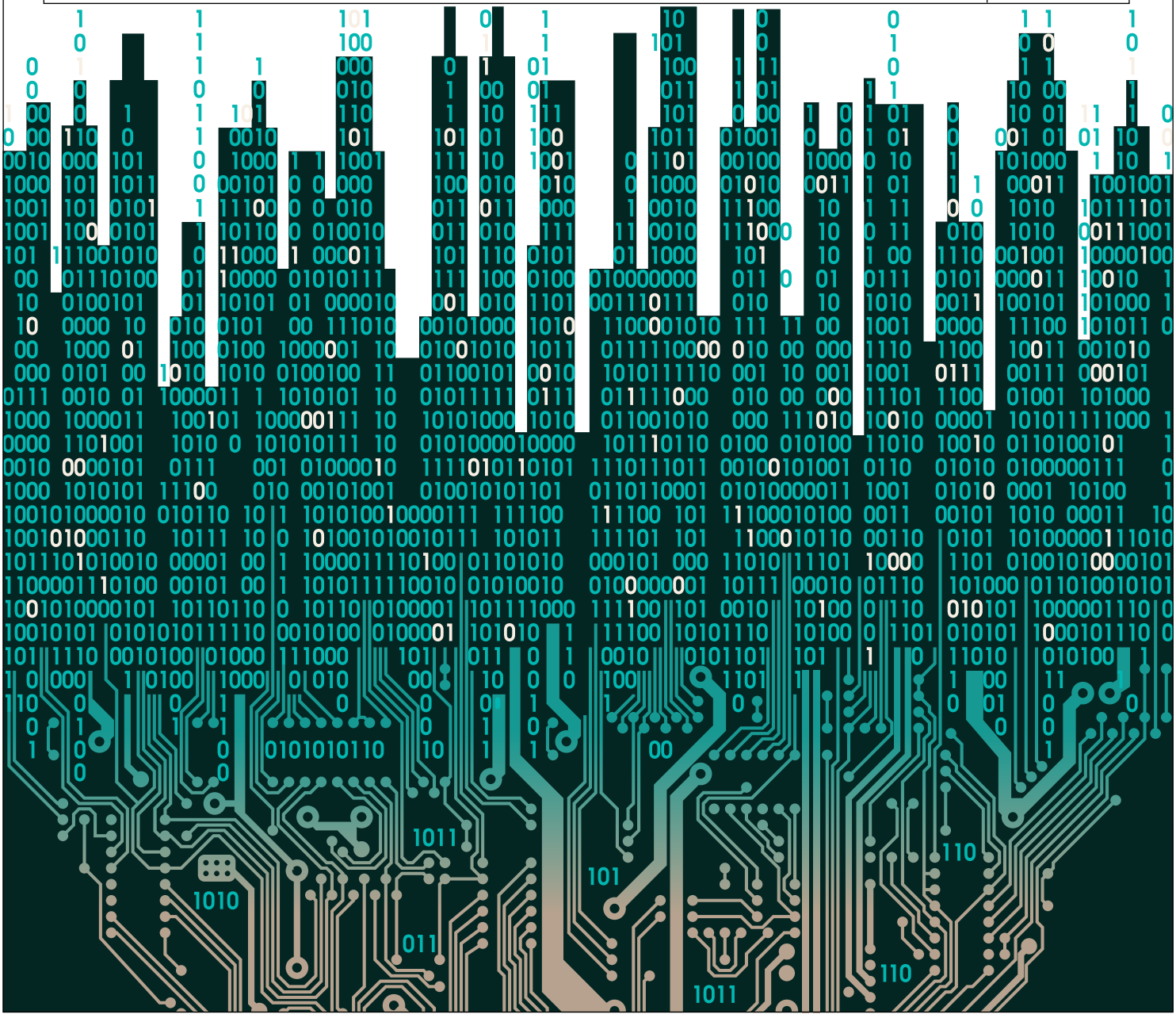
# Computer Science

## Sample Assesment Materials

**Pearson Edexcel Level 1/Level 2 GCSE (9-1) in Computer Science (1CP2)**

*First teaching September 2020*

*First certification 2022* | Issue 1



**Edexcel, BTEC and LCCI qualifications**

Edexcel, BTEC and LCCI qualifications are awarded by Pearson, the UK's largest awarding body offering academic and vocational qualifications that are globally recognised and benchmarked. For further information, please visit our qualification website at qualifications.pearson.com. Alternatively, you can get in touch with us using the details on our contact us page at qualifications.pearson.com/contactus

**About Pearson**

Pearson is the world's leading learning company, with 35,000 employees in more than 70 countries working to help people of all ages to make measurable progress in their lives through learning. We put the learner at the centre of everything we do, because wherever learning flourishes, so do people. Find out more about how we can help you and your learners at qualifications.pearson.com

*References to third party materials made in these sample assessment materials are made in good faith. Pearson does not endorse, approve or accept responsibility for the content of materials, which may be subject to change, or any opinions expressed therein. (Materials may include textbooks, journals, magazines and other publications and websites.)*

*All information in this document is correct at time of publication.*

ISBN 978 1 446 96580 1

All the material in this publication is copyright
© Pearson Education Limited 2020

# Contents


| | |
|---|---|
| Introduction | 1 |
| General marking guidance | 3 |
| Paper 1 | 5 |
| Paper 1 Mark scheme | 21 |
| Paper 2 | 37 |
| Programming Language Subset (PLS) | 45 |
| Paper 2 Mark scheme | 57 |

# Introduction

The Pearson Edexcel Level 1/Level 2 GCSE (9-1) in Computer Science is designed for use in schools and colleges. It is part of a suite of GCSE qualifications offered by Pearson.

These sample assessment materials have been developed to support this qualification and will be used as the benchmark to develop the assessments students will take.

# General marking guidance

- All candidates must receive the same treatment. Examiners must mark the last candidate in exactly the same way as they mark the first.
- Mark schemes should be applied positively. Candidates must be rewarded for what they have shown they can do, rather than be penalised for omissions.
- Examiners should mark according to the mark scheme – not according to their perception of where the grade boundaries may lie.
- All the marks on the mark scheme are designed to be awarded. Examiners should always award full marks if deserved, i.e. if the answer matches the mark scheme. Examiners should also be prepared to award zero marks if the candidate's response is not worthy of credit according to the mark scheme.
- Where some judgement is required, mark schemes will provide the principles by which marks will be awarded, and exemplification/indicative content will not be exhaustive. However, different examples of responses will be provided at standardisation.
- When examiners are in doubt regarding the application of the mark scheme to a candidate's response, a senior examiner must be consulted before a mark is given.
- Crossed-out work should be marked, unless the candidate has replaced it with an alternative response.

## How to award marks when level descriptions are used

### Finding the right level

The first stage is to decide which level the answer should be placed in. To do this, use a 'best-fit' approach, deciding which level most closely describes the quality of the answer. Answers can display characteristics from more than one level, and where this happens markers must use their professional judgement to decide which level is most appropriate.

### Placing a mark within a level

After a level has been decided on, the next stage is to decide on the mark within the level. The instructions below tell you how to reward responses within a level. However, where a level has specific guidance about how to place an answer within a level, always follow that guidance. Statements relating to the treatment of students who do not fully meet the requirements of the question are also shown in the indicative content section of each levels-based mark scheme. These statements should be considered alongside the level descriptors.

Markers should be prepared to use the full range of marks available in a level, and not restrict marks to the middle. Markers should start at the middle of the level (or the upper-middle mark if there is an even number of marks) and then move the mark up or down to find the best mark. To do this, they should take into account how far the answer meets the requirements of the level:

- If it meets the requirements fully, markers should be prepared to award full marks within the level. The top mark in the level is used for answers that are as good as can realistically be expected within that level
- If it only barely meets the requirements of the level, markers should consider awarding marks at the bottom of the level. The bottom mark in the level is used for answers that are the weakest that can be expected within that level

- The middle mark(s) of the level is used for answers that have a reasonable match to the descriptor. This might represent a balance between some characteristics of the level that are fully met and others that are only barely met.

Please check the examination details below before entering your candidate information

| Candidate surname | Other names |
|---|---|
| | |

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

Centre Number | | | | | |

Candidate Number | | | | |

**Sample assessment material for first teaching**
**September 2020**

Time: 1 hour 30 minutes | Paper Reference **1CP2/01**

# Computer Science

## Paper 1: Principles of Computer Science

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

Total Marks

### Instructions

- Use **black** ink or ball-point pen.
- **Fill in the boxes** at the top of this page with your name, centre number and candidate number.
- Answer **all** questions.
- Answer the questions in the spaces provided – *there may be more space than you need.*
- You are not allowed to use a calculator.

### Information

- The total mark for this paper is 75.
- The marks for **each** question are shown in brackets – *use this as a guide as to how much time to spend on each question.*

### Advice

- Read each question carefully before you start to answer it.
- Try to answer every question.
- Check your answers if you have time at the end.

*Turn over* ▶

S67285A

©2020 Pearson Education Ltd.

1/1/1

S 6 7 2 8 5 A 0 1 1 6

Pearson

**Answer ALL questions. Write your answers in the spaces provided.**

**Some questions must be answered with a cross in a box ☒. If you change your mind about an answer, put a line through the box ☒ and then mark your new answer with a cross ☒.**

**1 Data**

(a) Identify the smallest unit of measurement.

**(1)**

- ☒ **A** bit
- ☒ **B** byte
- ☒ **C** kibibyte
- ☒ **D** nibble

(b) Identify the maximum number of values that can be represented with 5 bits.

**(1)**

- ☒ **A** 5
- ☒ **B** 16
- ☒ **C** 25
- ☒ **D** 32

(c) A car park uses a number-plate recognition system.

(i) Identify the reason why an unsigned integer should be used to record the number of cars entering the car park, rather than a signed integer.

**(1)**

- ☒ **A** Unsigned integers are more accurate
- ☒ **B** Unsigned integers cannot have overflow errors
- ☒ **C** Unsigned integers store more positive values
- ☒ **D** Unsigned integers do not use a parity bit

(ii) The system stores images of car number plates.

Construct an expression to show how many bytes there are in 6 tebibytes.

You do not need to carry out the calculation.

**(3)**

.....

.....

.....

.....

DO NOT WRITE IN THIS AREA DO NOT WRITE IN THIS AREA DO NOT WRITE IN THIS AREA

(d) Identify the 4-bit binary addition that will result in an overflow error.

**(1)**

- ☒ **A** 1001+1000
- ☒ **B** 0011+1001
- ☒ **C** 1000+0110
- ☒ **D** 0111+1000

(e) Give the 8-bit binary representation of the denary number 82.

**(2)**

(f) (i) Convert the binary number 0011 1101 to hexadecimal.

**(2)**

(ii) Explain why hexadecimal notation is used.

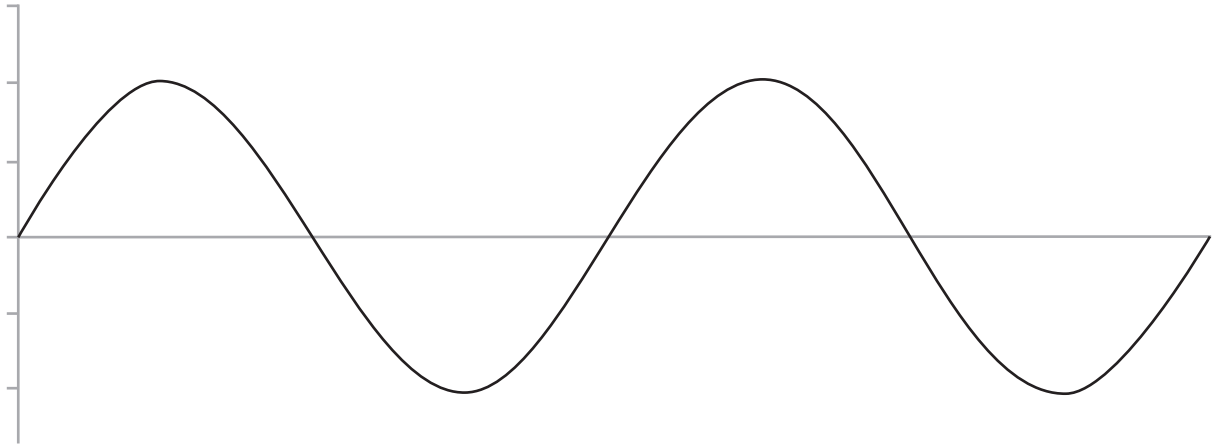
**(2)**

DO NOT WRITE IN THIS AREA
DO NOT WRITE IN THIS AREA
DO NOT WRITE IN THIS AREA

(g) An analogue to digital converter is used to change the sounds received by a microphone into a form that can be processed by a computer.

Complete the diagram to show a sample interval and label both axes.

**(3)**



(h) An image uses a 12-bit colour depth. It is 64 pixels wide and 48 pixels high.

Construct an expression to calculate the file size of the image in MiB.

You do not have to do the calculation.

**(4)**

**(Total for Question 1 = 20 marks)**

DO NOT WRITE IN THIS AREA
DO NOT WRITE IN THIS AREA
DO NOT WRITE IN THIS AREA

DO NOT WRITE IN THIS AREA
DO NOT WRITE IN THIS AREA
DO NOT WRITE IN THIS AREA

**2 Networks**

(a) A hotel chain has hotels in several countries and a head office in England.

State the type of network needed to connect these hotels to the head office.

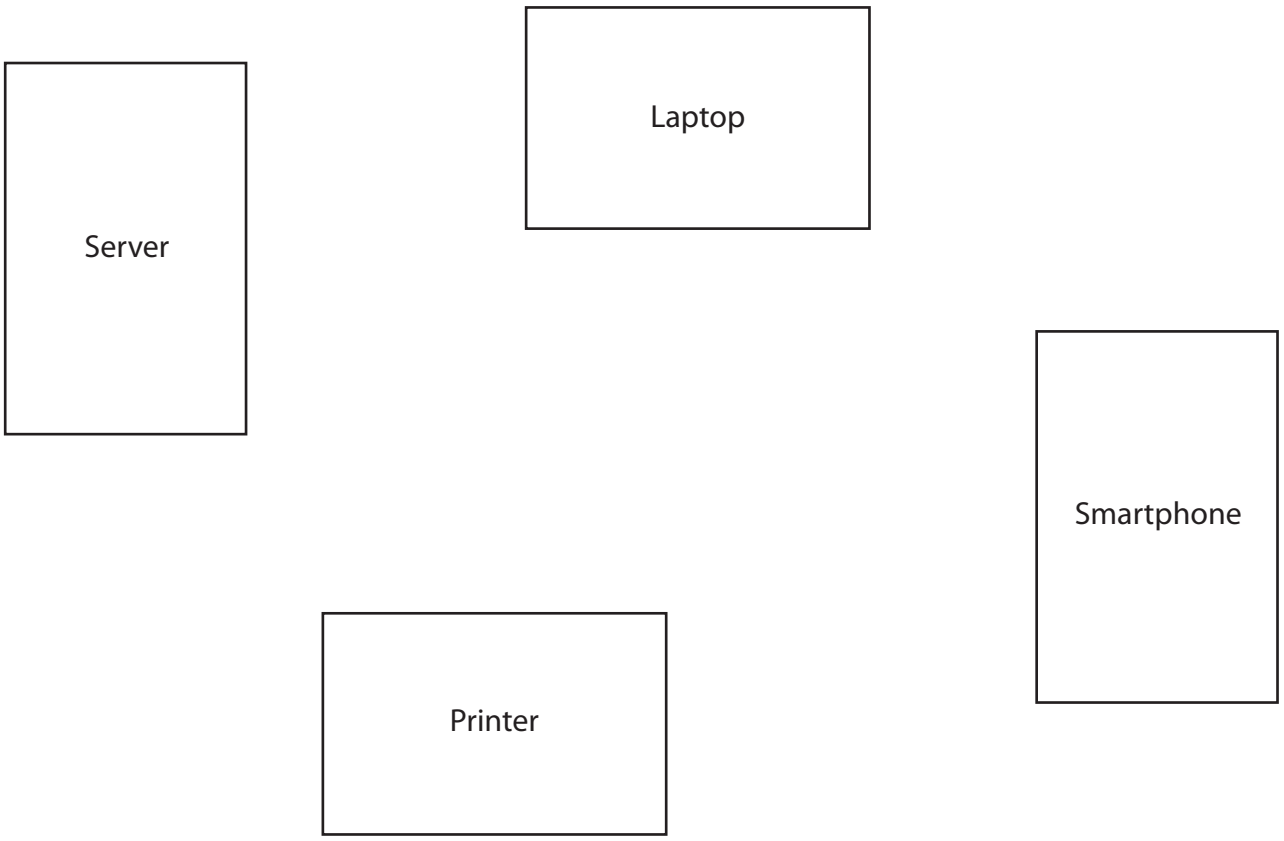
**(1)**

.....

.....

(b) Draw lines between these devices to show a fully-connected mesh network topology diagram:

**(2)**



(c) Routers send packets that contain data around the internet.

State **two other** items found in a packet.

**(2)**

1 .....

.....

2 .....

.....

(d) The transport layer of network protocols splits data into packets before sending it. All the packets are received correctly.

Describe the process that ensures the data received matches the original.

**(2)**

.....

.....

.....

.....

DO NOT WRITE IN THIS AREA

DO NOT WRITE IN THIS AREA

DO NOT WRITE IN THIS AREA

(e) Construct an expression to show how many seconds it will take to transmit 20 MiB of data using a network transmission speed of 2 Mbps.

You do not have to do the calculation.

**(4)**

**(Total for Question 2 = 11 marks)**

DO NOT WRITE IN THIS AREA
DO NOT WRITE IN THIS AREA
DO NOT WRITE IN THIS AREA

**3 Issues and impact**

(a) State **two** environmental issues associated with the disposal of digital technology.

**(2)**

1 .....

.....

2 .....

.....

(b) This notification appears on a computer screen.

**Thank you for clicking our link.**
Your important files are no longer accessible.

**Can I get access to my files?**
Yes, you can. Simply send your payment as described below.

**How long do I have?**
14 days.

**How do I pay?**
Send £500 in Bitcoin to abc123def456ghi789.

(i) Name the type of malware used in this cyberattack.

**(1)**

.....

.....

(ii) Explain **one** way that digital systems may be vulnerable to cyberattacks when users do not properly maintain their software.

**(2)**

.....

.....

.....

.....

DO NOT WRITE IN THIS AREA

DO NOT WRITE IN THIS AREA

DO NOT WRITE IN THIS AREA

(c) People use spoken commands to interact with digital assistants.

Explain **one** ethical concern associated with digital assistant technologies.

**(2)**

.....

.....

.....

.....

(d) Students are asked to sign an acceptable use policy.

Explain **one** way that an acceptable use policy helps to protect student data.

**(2)**

.....

.....

.....

.....

**(Total for Question 3 = 9 marks)**

DO NOT WRITE IN THIS AREA
DO NOT WRITE IN THIS AREA
DO NOT WRITE IN THIS AREA

**4 Computers**

(a) One function of utility software is to provide anti-malware.

State **two other** functions of utility software.

**(2)**

1 .....

.....

2 .....

.....

(b) State **one** way that a code review helps programmers to produce robust software.

**(1)**

.....

.....

(c) Describe the role of the control unit, the control bus, the data bus and the address bus when fetching an instruction from memory.

**(4)**

.....

.....

.....

.....

.....

.....

.....

.....

DO NOT WRITE IN THIS AREA

DO NOT WRITE IN THIS AREA

DO NOT WRITE IN THIS AREA

(d) The operating system controls the scheduling of processes.

Describe how the operating system uses scheduling to allocate processor time.

**(4)**

DO NOT WRITE IN THIS AREA

DO NOT WRITE IN THIS AREA

DO NOT WRITE IN THIS AREA

(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)**

DO NOT WRITE IN THIS AREA

DO NOT WRITE IN THIS AREA

DO NOT WRITE IN THIS AREA

**(Total for Question 4 = 17 marks)**

DO NOT WRITE IN THIS AREA
DO NOT WRITE IN THIS AREA
DO NOT WRITE IN THIS AREA

**5 Computational thinking**

(a) State the type of error that can be found in **algorithms**.

**(1)**

.....

.....

(b) A binary search algorithm is used with this list to find the target value 'b'.

a b c d e f g h i j k

Complete the table to show the **three** characters in the order that the algorithm would compare them against the target value.

**(3)**

| | |
|---|---|
| First | |
| Second | |
| Third | |
| Fourth | b |

(c) Programmers can use all capitals to show that a value is a constant.

A constant is shown here on line 3.

```
2  # Prototype for the main swimming pool
3  MAX_CAPACITY = 120       # Maximum number of swimmers
4
5  numAdult = 14            # Current number of adults swimming
6  numChild = 73            # Current number of children swimming
```

Explain **one** reason why programmers use signals indicating a value is a constant, rather than repeating the same fixed value throughout an algorithm.

**(2)**

.....

.....

.....

.....

DO NOT WRITE IN THIS AREA

DO NOT WRITE IN THIS AREA

DO NOT WRITE IN THIS AREA

(d) Margaret owns an ice-cream shop.

This program manipulates sales figures from Margaret's shop.

```
num = 0
x = 999
y = 0
line = ""

f = open("SalesFile.txt", "r")
for line in f:
    num = int(line)
    if num < x:
        x = num
    if num > y:
        y = num
print(x, y)
f.close()
```

The only inputs from the file to the program are 355, 554, 199 and 409.

Complete the trace table showing the execution of the program with these four inputs.

You may not need to fill in all the rows in the table.

**(6)**

| num | x | y | Display |
|---|---|---|---|
| | | | |
| | | | |
| | | | |
| | | | |
| | | | |
| | | | |
| | | | |
| | | | |
| | | | |
| | | | |

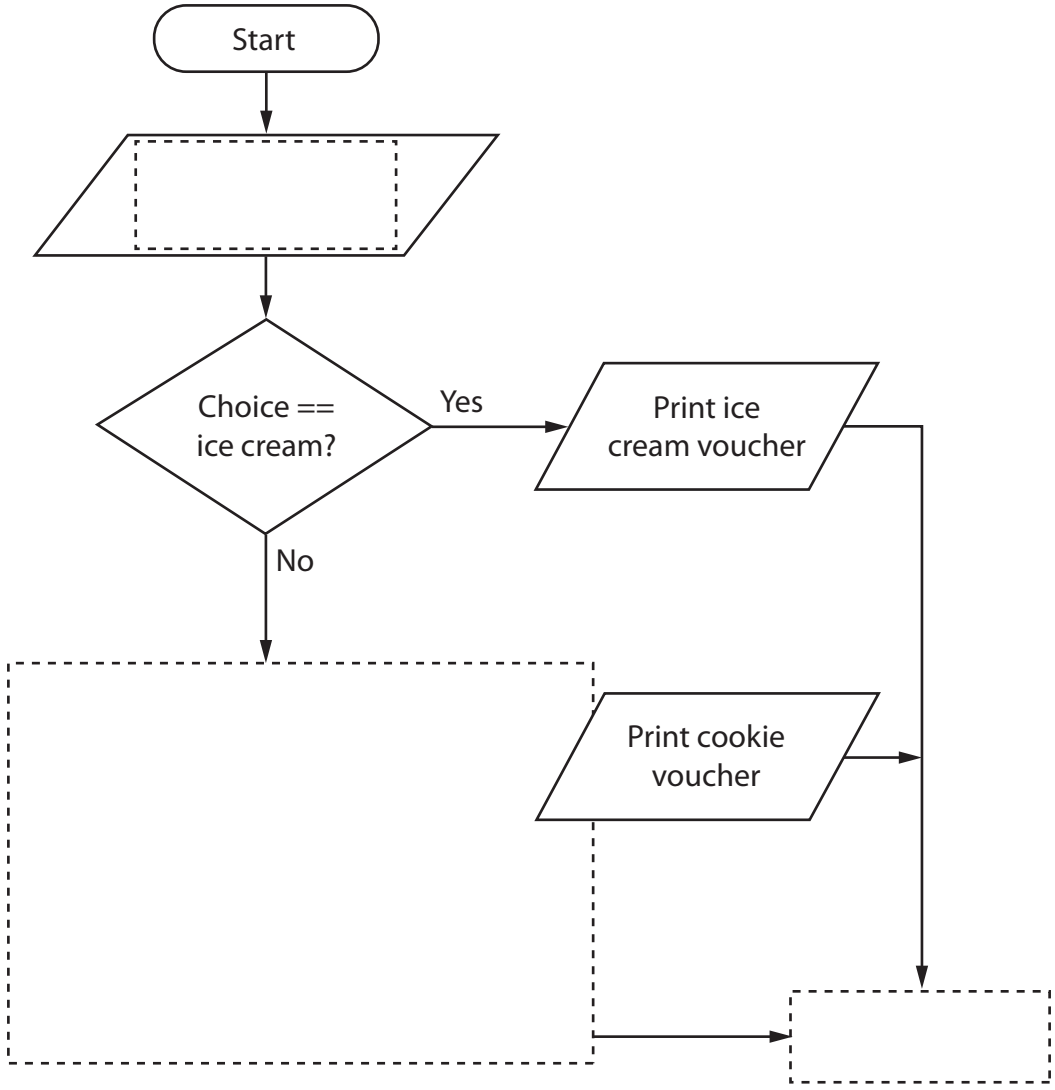
DO NOT WRITE IN THIS AREA
DO NOT WRITE IN THIS AREA
DO NOT WRITE IN THIS AREA

(e) Customers can get a voucher for their favourite item.

- Customers whose favourite item is ice cream get a voucher for ice cream.
- Customers whose favourite item is cookies get a voucher for cookies.
- Customers who do not choose either get a voucher for drinks.

Complete the flowchart to show this process.

**(6)**




**(Total for Question 5 = 18 marks)**

**TOTAL FOR PAPER = 75 MARKS**

DO NOT WRITE IN THIS AREA
DO NOT WRITE IN THIS AREA
DO NOT WRITE IN THIS AREA

## Paper 1 mark scheme

| Question number | Answer | Additional guidance | Mark |
|---|---|---|---|
| **1(a)** | **A** bit<br><br>***B*** *is not correct because a bit is smaller than a byte*<br><br>***C*** *is not correct because a bit is smaller than a kibibyte*<br><br>***D*** *is not correct because a bit is smaller than a nibble* | | **(1)** |

| Question number | Answer | Additional guidance | Mark |
|---|---|---|---|
| **1(b)** | **D** 32<br><br>***A*** *is not correct because 5 bits can represent 32 values*<br><br>***B*** *is not correct because 5 bits can represent 32 values*<br><br>***C*** *is not correct because 5 bits can represent 32 values* | | **(1)** |

| Question number | Answer | Additional guidance | Mark |
|---|---|---|---|
| **1(c)(i)** | **C** Unsigned integers store more positive values<br><br>***A*** *is not correct because unsigned integers are not more accurate*<br><br>***B*** *is not correct because overflow errors can still occur with unsigned integers*<br><br>***D*** *is not correct because the use of a parity bit is not relevant to the scenario* | | **(1)** |

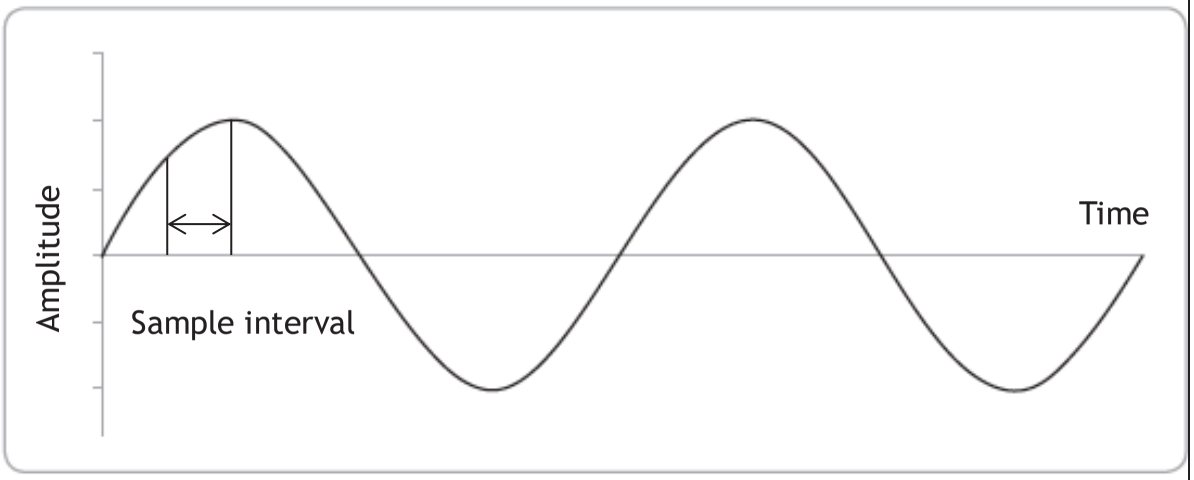
| Question number | Answer | Additional guidance | Mark |
|---|---|---|---|
| **1(c)(ii)** | 6 x $1024^4$<br><br>Award **1** mark for sight of 1024<br><br>Award **1** mark for sight of $^4$ (applied only to 1024)<br><br>Award **1** mark for sight of x6<br><br>Award **all** marks if the result of the calculation is given: 6,597,069,766,656 (bytes).<br>**N.B.** This is not needed or expected. | Equivalent expressions are awarded. | **(3)** |

| Question number | Answer | Additional guidance | Mark |
|---|---|---|---|
| **1(d)** | **A** 1001+1000<br><br>***B*** *is not correct because it will result in 1100*<br><br>***C*** *is not correct because it will result in 1110*<br><br>***D*** *is not correct because it will result in 1111* | | **(1)** |

| Question number | Answer | Additional guidance | Mark |
|---|---|---|---|
| **1(e)** | 0101 0010 | Award 1 mark for each nibble in the correct location. | **(2)** |

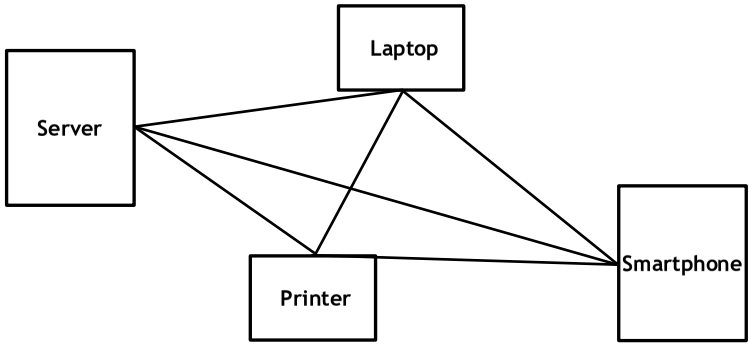
| Question number | Answer | Additional guidance | Mark |
|---|---|---|---|
| **1(f)(i)** | 3D | Award 1 mark for each nibble in the correct location. | **(2)** |

| Question number | Answer | Additional guidance | Mark |
|---|---|---|---|
| **1(f)(ii)** | Award **1** mark for the identification of the reason (1) with a linked justification/exemplification (1), up to a maximum of **2** marks.<br><br>Hexadecimal is used as shorthand for binary / uses fewer digits/characters (1), so **humans** make fewer mistakes / find it easier to read/understand/remember/manipulate (1). | Do not accept answers suggesting that fewer digits save storage/memory. | **(2)** |

| Question number | Answer | Additional guidance | Mark |
|---|---|---|---|
| **1(g)** | X axis correctly labelled (1)<br>Y axis correctly labelled (1)<br><br>Sample interval indicated (1)<br>Do not award marks if wavelength is labelled rather than sample interval.<br>Accept for sample interval if two points given that are shorter than the wavelength.<br>  <br> | Accept a unit of time for the X axis.<br><br>Accept signal strength for the Y axis. | **(3)** |

| Question number | Answer | Additional guidance | Mark |
|---|---|---|---|
| **1(h)** | Award **1** mark for:<br>Sight of: 64 × 48 x 12<br><br>Award **1** mark for:<br>Sight of: 1024 x 8<br><br>Award **1** mark for:<br>Sight of: (1024) × (1024) **OR** $(1024)^2$<br><br>Award **1** mark for correct numerator/denominator orientation.<br><br>Examples of expressions that gains full marks:<br>$\frac{64 \times 48 \times 12}{1024 \times 1024 \times 8}$ | Accept any other equivalent mathematical expression. | **(4)** |

| Question number | Answer | Additional guidance | Mark |
|---|---|---|---|
| **2(a)** | WAN/Wide Area Network | | **(1)** |

| Question number | Answer | Additional guidance | Mark |
|---|---|---|---|
| **2(b)** |  <br>• **1** mark for at least two connections to each device.<br>• **2** marks for three connections to each device. | | **(2)** |

| Question number | Answer | Additional guidance | Mark |
|---|---|---|---|
| **2(c)** | Any **two** items from:<br><br>• Destination address (1)<br>• Source address (1)<br>• Error checking field / check sum (1)<br>• Time stamp (1)<br>• Sequence number (1).<br><br>Accept any other appropriate response. | | **(2)** |

| Question number | Answer | Additional guidance | Mark |
|---|---|---|---|
| **2(d)** | A linked description that makes reference to the following points:<br><br>Each packet has a sequence number (added at the sending end) (1)<br>The packets are put back into (sequence) order (at the destination) (1). | | **(2)** |

| Question number | Answer | Additional guidance | Mark |
|---|---|---|---|
| **2(e)** | Total number of bits to transfer:<br>**1** mark for $20 \times 1024^2$<br>**1** mark for $\times 8$<br><br>Speed in bits per second:<br>**1** mark for $2 \times 1000000$<br><br>Numerator/denominator:<br>**1** mark for<br>$$\frac{bits\ to\ transfer}{bits\ per\ second}$$<br><br>e.g.:<br>$$\frac{20 \times 1024 \times 1024 \times 8}{2 \times 1000000}$$ | Any equivalent expression to be awarded. | **(4)** |

| Question number | Answer | Additional guidance | Mark |
|---|---|---|---|
| **3(a)** | Any **two** issues from:<br><br>• Waste materials end in landfill (1)<br>• Dangerous toxins are released into the ground/water (1)<br>• Waste is transported overseas (carbon emissions) (1). | | **(2)** |

| Question number | Answer | Additional guidance | Mark |
|---|---|---|---|
| **3(b)(i)** | Ransomware | | **(1)** |

| Question number | Answer | Additional guidance | Mark |
|---|---|---|---|
| **3(b)(ii)** | Award **1** mark for the identification of a way (1) with a linked justification/exemplification (1), up to a maximum of **2** marks.<br><br>• Software may contain security bugs (1) because it is unpatched (1).<br>• Anti-malware may not identify an attack (1) because the virus definitions are out of date (1).<br><br>Accept any other appropriate response. | | **(2)** |

| Question number | Answer | Additional guidance | Mark |
|---|---|---|---|
| **3(c)** | Award **1** mark for the identification of an ethical concern (1) with a linked justification/exemplification (1), up to a maximum of **2** marks.<br><br>• The data may no longer be private (1) because companies may share it (1).<br>• People may not realise their data is analysed (1) because it is unclear who owns the data (1).<br>• People are willing to sacrifice (some) privacy (1) in return for access to services (1).<br><br>Accept any other appropriate response. | | **(2)** |

| Question number | Answer | Additional guidance | Mark |
|---|---|---|---|
| **3(d)** | Award **1** mark for the identification of a way (1) with a linked justification/exemplification (1), up to a maximum of **2** marks.<br><br>• Students are deterred from unsafe practices (1) because consequences are clarified (1).<br>• People follow safe/good practices (1) because permitted activities are defined/set out (1).<br><br>Accept any other appropriate response. | | **(2)** |

| Question number | Answer | Additional guidance | Mark |
|---|---|---|---|
| **4(a)** | Any **two** functions from:<br><br>• Repairing files (1)<br>• Compression (1)<br>• Defragmentation (1)<br>• Back-up (1)<br>• Firewall (1)<br>• Managing application updates (1)<br>• Formatting disks/drives (1)<br>• System analysis tools (1). | | **(2)** |

| Question number | Answer | Additional guidance | Mark |
|---|---|---|---|
| **4(b)** | Any **one** way from:<br><br>• By identifying bad programming practices (1)<br>• By identifying vulnerabilities in the code (1)<br>• By checking for efficiency of code (1). | Do not accept 'Checking for errors' if not qualified with 'not picked up in testing'. | **(1)** |

| Question number | Answer | Additional guidance | Mark |
|---|---|---|---|
| **4(c)** | A linked description that makes reference to the **four** following points:<br><br>• The address of memory (holding instruction) is placed on the address bus (1).<br>• The control unit sends a signal (1) on the control bus (to start a read operation) (1).<br>• The instruction is/the contents of the memory are placed on the data bus (1). | | **(4)** |

| Question number | Answer | Additional guidance | Mark |
|---|---|---|---|
| **4(d)** | A linked description that makes reference to any **four** of the following points:<br><br>• All processes are held in a queue (1)<br>• Processes are prioritised (1)<br>• Processes are allocated time slices (1)<br>• Length of time slice depends on priority (1)<br>• (and) processes are switched (at the end of their time slice) (1)<br>• Unfinished processes are put to the back of the queue (1)<br>• During the time slice the process has exclusive use of the processor (1). | | **(4)** |

| Question number | Indicative content | Mark |
|---|---|---|
| **4(e)** | Advantages of high-level languages:<br>• 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.<br>• 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.<br>• 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 relatively quickly.<br>• High-level languages use keywords, which will enable team members to read and understand the code for the user interface more easily.<br>• There are lots of people who can program in a high-level language, making it relatively easy to recruit experienced programmers to the user interface team.<br>• As high-level language translators exist for a range of operating systems, each member of the user interface team can develop code in their preferred environment.<br><br>Advantages of low-level languages:<br>• Code written in assembly language normally executes more quickly and takes up less memory than code written in a high-level language. This may be crucial to enable the control unit for the alarm system to function effectively.<br>• There may be no high-level language for the microprocessor chip inside the control unit, so an assembly language would have to be used for it.<br>• Code written in assembly language allows the programmer to directly control system hardware. | **(6)** |

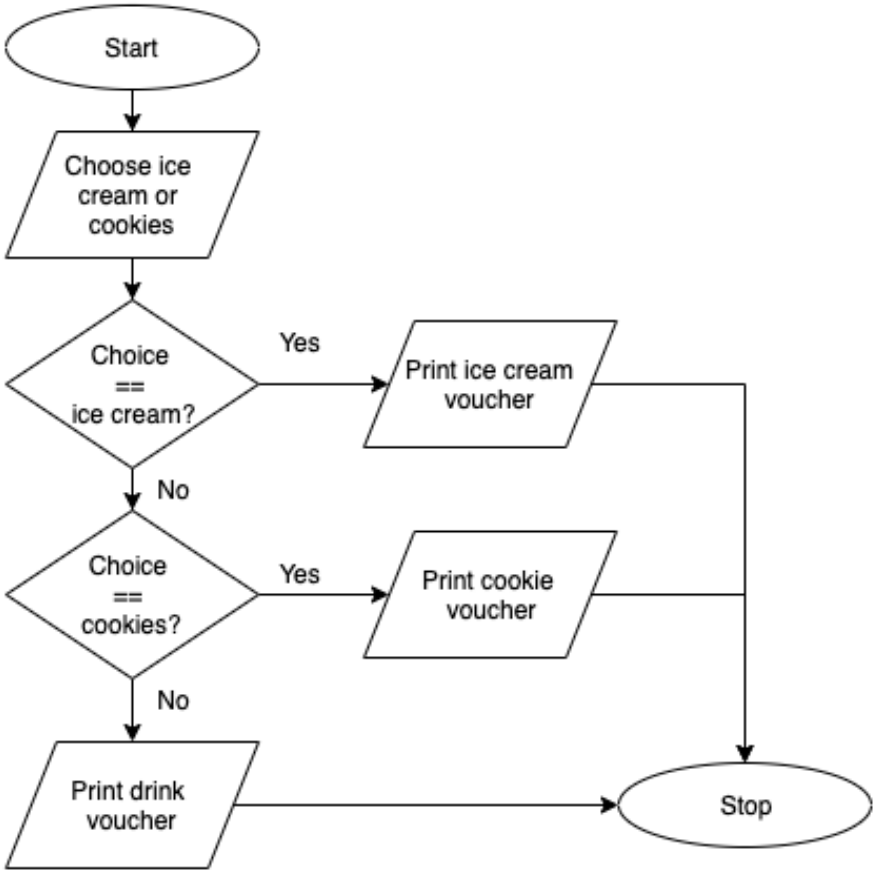
| Level | Mark | Descriptor |
|---|---|---|
| | 0 | No rewardable content. |
| Level 1 | 1–2 | Basic, independent points are made, showing elements of understanding of key concepts/principles of computer science. (AO1)<br><br>The discussion will contain basic information with little linkage between points made or application to the context. (AO2) |
| Level 2 | 3–4 | Demonstrates adequate understanding of key concepts/principles of computer science. (AO1)<br><br>The discussion shows some linkages and lines of reasoning with some structure and application to the context. (AO2) |
| Level 3 | 5–6 | Demonstrates comprehensive understanding of key concepts/principles of computer science to support the discussion being presented. (AO1)<br><br>The discussion is well developed, with sustained lines of reasoning that are coherent and logically structured, and which clearly apply to the context. (AO2) |

| Question number | Answer | Additional guidance | Mark |
|---|---|---|---|
| **5(a)** | Logic | | **(1)** |

| Question number | Answer | Additional guidance | Mark |
|---|---|---|---|
| **5(b)** | Award **1** mark for each correct cell.<br><br>First: f<br>Second: c<br>Third: a<br>Fourth: b | | **(3)** |

| Question number | Answer | Additional guidance | Mark |
|---|---|---|---|
| **5(c)** | Award **1** mark for the identification of a reason (1) with a linked justification/exemplification (1), up to a maximum of **2** marks.<br><br>• Constants (shown in all capitals) are less likely to be changed by accident or error (1), so algorithms that use them should be more robust (1).<br>• If the value of a constant does have to be altered (1), only one change is required (on the line where it is created and set) (1).<br>• Constants allow values to be replaced with a name/identifier (1), so code is easier to read/maintain (1). | | **(2)** |

<table>
<tr><th>Question number</th><th>Answer</th><th>Additional guidance</th><th>Mark</th></tr>
<tr><td>5(d)</td><td>1 mark for initialising all variables and 1 mark for each correct pass through the loop.
<table>
<tr><th>num</th><th>x</th><th>y</th><th>Display</th><th>Marks</th></tr>
<tr><td>0</td><td>999</td><td>0</td><td></td><td>(1)</td></tr>
<tr><td>355</td><td>355</td><td>355</td><td></td><td>(1)</td></tr>
<tr><td>554</td><td></td><td>554</td><td></td><td>(1)</td></tr>
<tr><td>199</td><td>199</td><td></td><td></td><td>(1)</td></tr>
<tr><td>409</td><td></td><td></td><td></td><td>(1)</td></tr>
<tr><td></td><td></td><td></td><td>199 554</td><td>(1)</td></tr>
</table></td><td>• Award alternative versions of the trace table if correct. For example, copying of values that do not change.<br>• Passes are incorrect if display is indicated.<br>• Display must be after the final pass (on a separate line in the table).</td><td>(6)</td></tr>
</table>

| Question number | Answer | Additional guidance | Mark |
|---|---|---|---|
| **5(e)** | • Correct message in output box acting as a prompt for the user (1).<br>• Correct diamond symbol for decision (1).<br>• Correct test 'Choice == cookies?' for decision (1).<br>• Correct label 'Yes' on right arrow AND Correct label 'No' on bottom arrow (1).<br>• Correct output symbol with suitable message (1).<br>• Correct ellipse symbol and 'stop' for terminator (1).<br>  | • Symbol and contents are awarded independently.<br>• Award 'End', 'Stop', 'Start' and 'Begin' as text for terminator symbols.<br>• Award '==' and '=' used for equivalence inside decision symbol, but not in process symbol.<br>• Accept 'Input choice' as an alternative response in the top process symbol | **(6)** |

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

**Sample assessment material for first teaching September 2020**

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 which shows line numbers
- a 'STUDENT CODING' folder containing code and data files
- printed and electronic copies of the Programming Language Subset (PLS) document.

### Instructions

- Answer all the questions on your computer.
- Save new or amended code using the file name provided and place it in the 'COMPLETED CODING' folder.
- You must **not** use the internet at any time during the examination.

### Information

- The 'STUDENT CODING' folder in your user area includes all the code and data files you need.
- The total mark for this paper is 75.
- The marks for each question are shown in brackets.

### Advice

- Read each question carefully before you start to answer it.
- Save your work regularly.
- Check your answers and work if you have time at the end.

*Turn over* ▶

S67286A

©2020 Pearson Education Ltd.

1/1/1/1

S 6 7 2 8 6 A

Pearson

**Answer ALL questions.**

**Suggested time: 10 minutes**

**1** A program simulates the roll of a dice. The program uses a random number generator to create a random integer, between 1 and 6, to represent the roll.

Open file **Q01**.

Amend the code to add or complete lines to:

- import the random library
- create one variable
- create one constant
- assign the result of a library call to a variable
- display a message and the contents of a variable on the screen.

Do **not** add any additional functionality.

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

**(Total for Question 1 = 7 marks)**

**Suggested time: 20 minutes**

**2** A programmer has started to write a program, but it does not work correctly. The program should ask the user, "Would you like me to sing?". The user then responds 'y' or 'n'. If the user types 'y', then the computer displays the lines of a song.

Open file **Q02**.

Amend the code to:

- fix the syntax error on line 13
- fix the syntax error on line 15
- fix the syntax error on line 17
- change the identifier x to a more meaningful name
- display a suitable question for the user
- accept the user's input of 'y' or 'n' (no validation is required)
- add a comment to explain the effect of the range function's last parameter, set to –1
- add two uses of white space to aid readability.

Do **not** add any additional functionality.

Save your amended code file as **Q02FINISHED.py**

**(Total for Question 2 = 10 marks)**

**Suggested time: 20 minutes**

**3** A program is needed that must meet the following requirements:

- accept only numbers from 1 to 20, inclusive
- print 'invalid input' if the input is outside the permitted range. You can assume only numbers will be entered
- print the input number followed by 'is even', if the number is even
- print the input number followed by 'is odd', if the number is odd.

Open file **Q03**.

Amend the code to:

- fix the runtime error caused by inputting a valid number such as 4 (no other validation is required)
- fix the logic errors that cause incorrect or no outputs for acceptable numbers from 1 to 20, inclusive
- fix the logic error that causes the program to execute, even if a number outside the permitted range is entered
- make all output messages match the requirements listed above.

Do **not** add any additional functionality.

Save your amended code file as **Q03FINISHED.py**

**(Total for Question 3 = 13 marks)**

**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)**

**Suggested time: 25 minutes**

**5** Students are collecting data about the amount of water needed to fill different sized paper cones. Their measurements are compared to a calculated volume.

The formula to calculate the volume of a cone is:

$$V = \frac{1}{3}\pi r^2 h$$

- $V$ is volume
- $\pi$ is the constant Pi
- $r$ is the radius of the base of the cone
- $h$ is the height of the cone.

A program and subprogram have been started to carry out the calculation.

Open file **Q05**.

Amend the program and subprogram to meet the following requirements:

- the subprogram must work for any values of radius and height passed as parameters. You can assume values passed to the subprogram will always be numbers. No validation is required
- the subprogram must calculate the volume based on the input parameters
- the main program must print the volume, formatted to show three decimal places (e.g. 16.135).

Do **not** add any additional functionality.

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

**(Total for Question 5 = 15 marks)**

**Suggested time: 25 minutes**

**6** A program is needed to authenticate system logins consisting of a login name and a four-digit passcode.

These items for current users are stored in a two-dimensional list with 19 records. The list contains user number, last name, first name, login name and passcode.

The list is sorted by login name. All users have passcodes that are integers between 1000 and 9999.

Open file **Q06**.

Write a program to meet the following requirements:

**Inputs**

- Prompt for and accept a login name, no validation required
- Prompt for and accept a four-digit passcode. You can assume only numbers will be entered
- Ensure the passcode is between 1000 and 9999, inclusive

**Process**

- Work with any number of users in the list
- Use a linear search to find the record with the correct combination of login name and passcode
- Stop searching when the location of where the record should have been found is passed. For example:
  - if looking for 'Jam118' and find 'Joy116', then the record is not in the list, so the search can stop
- Does not need to loop continuously

**Outputs**

- Display a welcome message, including the user's first and last names, if the login name and passcode are found
- Display an invalid input message, if the login name and passcode are not found.

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)**

**TOTAL FOR PAPER = 75 MARKS**

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

**Sample assessment material for first teaching September 2020**

Paper Reference **1CP1/02**

# Computer Science

## Programming Language Subset
## Issue 1

**PLS Booklet**

**Do not return this booklet with the question paper.**

*Turn over* ▶

S67286A

©2020 Pearson Education Ltd.

1/1/1/1

S 6 7 2 8 6 A

Pearson

**Introduction**

The Programming Language Subset (PLS) is a document that specifies which parts of Python 3 are required in order that the assessments can be undertaken with confidence. Students familiar with everything in this document will be able to access all parts of the Paper 2 assessment. This does not stop a teacher/student from going beyond the scope of the PLS into techniques and approaches that they may consider to be more efficient or engaging.

Pearson will **not** go beyond the scope of the PLS when setting assessment tasks. Any student successfully using more esoteric or complex constructs or approaches not included in this document will still be awarded marks in Paper 2 if the solution is valid.

The pair of <> symbols indicate where expressions or values need to be supplied. They are not part of the PLS.

**Comments**

Anything on a line after the character # is considered a comment.

**Identifiers**

Identifiers are any sequence of letters, digits and underscores, starting with a letter. Both upper and lower case are supported.

**Data types and conversion**

**Primitive data types**

Variables may be explicitly assigned a data type during declaration.

Variables may be implicitly assigned a data type during initialisation.

Supported data types are:

| Data type | PLS |
|---|---|
| integer | int |
| real | float |
| Boolean | bool |
| character | str |

**Conversion**

Conversion is used to transform the data types of the contents of a variable using int(), str(), float(), bool() or list(). Conversion between any allowable types is permitted.

**Constants**

Constants are conventionally named in all upper-case characters.

**Combining declaration and initialisation**

The data type of a variable is implied when a variable is assigned a value.

**Structured data types**

A structured data type is a sequence of items, which themselves are typed. Sequences start with an index of zero.

| Data type | Explanation | PLS |
|---|---|---|
| string | A sequence of characters | str |
| array | A sequence of items with the same (homogeneous) data type | list |
| record | A sequence of items, usually of mixed (heterogenous) data types | list |

**Dimensions**

The number of dimensions supported by the PLS is two.

The PLS does not support ragged data structures. Therefore, in a list of records each record will have the same number of fields.

**Operators**

**Arithmetic operators**

| Arithmetic operator | Operation |
|---|---|
| / | division |
| * | multiplication |
| ** | exponentiation |
| + | addition |
| - | subtraction |
| // | integer division |
| % | modulus |

**Relational operators**

| Operator | Operator meaning |
|---|---|
| == | equal to |
| != | not equal to |
| > | greater than |
| >= | greater than or equal to |
| < | less than |
| <= | less than or equal to |

**Boolean operators**

| Operator |
|---|
| and |
| or |
| not |

**Programming constructs**

**Assignment**

Assignment is used to set or change the value of a variable.

| |
|---|
| `<variable identifier> = <value>` |
| `<variable identifier> = <expression>` |

**Sequence**

Every instruction comes one after the other, from the top of the file to the bottom of the file.

**Blocking**

Blocking of code segments is indicated by indentation and subprogram calls. These determine the scope and extent of variables they declare.

**Selection**

| | |
|---|---|
| `if <expression>:`<br>`  <command>` | If <expression> is true, then command is executed. |
| `if <expression>:`<br>`  <command>`<br>`else:`<br>`  <command>` | If <expression> is true, then first <command> is executed, otherwise second <command> is executed. |
| `if <expression>:`<br>`  <command>`<br>`elif <expression>:`<br>`  <command>`<br>`else:`<br>`  <command>` | If <expression> is true, then first <command> is executed, otherwise the second <expression> test is checked. If true, then second <command> is executed, otherwise third <command> is executed.<br>Supports multiple instances of 'elif'.<br>The 'else' is not required with the 'elif'. |

**Repetition**

| | |
|---|---|
| `while <condition>:`<br>`  <command>` | Pre-conditioned loop. This executes <command> while <condition> is true. |

**Iteration**

| | |
|---|---|
| `for <id> in <structure>:`<br>`  <command>` | Executes <command> for each element of a data structure, in one dimension. |
| `for <id> in range (<start>, <stop>):`<br>`  <command>` | Count controlled loop. Executes <command> a fixed number of times, based on the numbers generated by the range function. |
| `for <id> in range (<start>, <stop>, <step>):`<br>`  <command>` | Same as above, except that <step> influences the numbers generated by the range function. |

**Subprograms**

| | |
|---|---|
| `def <procname>():`<br>`  <command>` | A procedure with no parameters |
| `def <procname>(<paramA>, <paramB>):`<br>`  <command>` | A procedure with parameters |
| `def <funcname>():`<br>`  <command>`<br>`  return (<value>)` | A function with no parameters |
| `def <funcname>(<paramA>, <paramB>):`<br>`  <command>`<br>`  return (<value>)` | A function with parameters |

**Inputs and outputs**

**Screen and keyboard**

| | |
|---|---|
| `print (<item>)` | Displays <item> on the screen |
| `input (<prompt>)` | Displays <prompt> on the screen and returns the line typed in |

**Files**

The PLS supports manipulation of comma separated value text files.

File operations include open, close, read, write and append.

| | |
|---|---|
| `<fileid> = open(<filename>, "r")` | Opens file for reading |
| `for <line> in <fileid>:` | Reads every line, one at a time |
| `<alist> = <fileid>.readlines()` | Returns a list where each item is a line from the file |
| `<aline> = <fileid>.readline()` | Returns a line from a file. Returns an empty string on the end of the file |
| `<fileid> = open(<filename>, "w")` | Opens file for writing |
| `<fileid> = open(<filename>, "a")` | Opens file for appending |
| `<fileid>.writelines(<structure>)` | Writes <structure> to a file. <structure> is a list of strings |
| `<fileid>.write(<aString>)` | Writes a single string to a file |
| `<fileid>.close()` | Closes file |

**Library modules**

**Built-in functions**

The PLS supports these built-in functions by default. No import statements are necessary.

| Function | Description |
|---|---|
| `chr(integer)` | Returns the string which matches the Unicode value of integer. |
| `input(prompt)` | Displays the content of prompt to the screen and waits for the user to type in characters followed by a new line. |
| `len(object)` | Returns the length of the object, such as a string, one-dimensional or two-dimensional data structure. |
| `ord(char)` | Returns the integer equivalent to the Unicode string of a single 'char'. |
| `print(item)` | Prints item to the display. |
| `range(start, stop, step)` | The range() function is used to generate a list of numbers using step, beginning with the first and up to, but not including, the last. |
| `round(x, n)` | Rounds x to the number of n digits after the decimal (uses the 0.5 rule). |

**List methods**

No import necessary.

The supported functionality is as follows:

| Function | Description |
|---|---|
| `list.append(item)` | Adds an item to the end of the list |
| `del <list>[<index>]` | Removes the item at index from list |
| `list.insert(index, item)` | Inserts an item just before an existing one at index |
| `<item> = list()`<br>`<item> = []` | Two methods of creating a list structure. Both are empty |

**Random module**

Import the random module.

The supported functionality is as follows:

| Function | Description |
|---|---|
| `random.randint(a,b)` | Returns a random integer X so that a <= X <= b |
| `random.random()` | Returns a float number in the range of 0.0 and 1.0 |

**String module**

No import necessary.

The supported functionality is as follows:

| Function | Description |
|---|---|
| `len(<string>)` | Returns the length of <string> |
| `find(<string>)` | Returns the location of <string> in the original. Returns -1, if not found |
| `index(<string>)` | Returns the location of <string> in the original. Raises an exception if not found |
| `isalpha(<string>)` | Returns True, if all characters are alphabetic (a–z) |
| `isalnum(<string>)` | Returns True, if all characters are alphabetic (a–z) and digits (0–9) |
| `isdigit(<string>)` | Returns True, if all characters are digits (0–9), exponents are digits |
| `isnumeric(<string>)` | Returns True, if all characters are numeric (0–9), exponents and fractions are numeric |
| `replace(s1,s2)` | Returns original string with all occurrences of s1 replaced with s2 |
| `split(<char>)` | Returns a list of all substrings in the original, using <char> as the separator |
| `strip(<char>)` | Returns original string with all occurrences of 'char' removed from the front and back |
| `upper(<string>)` | Returns the original string in upper case |
| `lower(<string>)` | Returns the original string in lower case |
| `isupper(<string>)` | Returns True, if all characters are upper case |
| `islower(<string>)` | Returns True, if all characters are lower case |

Concatenation of strings is done using the + operator.

**Formatting strings**

Output can be customised to suit the problem requirements and the user's needs by using the `string.format()` method, with positional parameters.

Here is an example:

```
"{:>10} string, {:^5d} integer, {:7.4f} a real".format("Fred", 358,
3.14159)
```

| Category | Description |
|---|---|
| `Numbers` | • Decimal integer (d)<br>• Fixed point (f) |
| `Alignment` | • Left (<)<br>• Right (>)<br>• Centred (^)<br>• Repeat (*) |
| `Field size` | The total width of a field, regardless of how many columns are occupied |

**Math module**

Import the math module.

The supported functionality is as follows:

| Category | Description |
|---|---|
| `math.ceil(r)` | Returns the smallest integer not less than r |
| `math.floor(r)` | Returns the largest integer not greater than r |
| `math.sqrt(x)` | Returns the square root of x |
| `math.pi` | The constant Pi (∏) |

**Time module**

Import the time module.

The supported functionality is as follows:

| Function | Description |
|---|---|
| `time.sleep(sec)` | The current process is suspended for the given number of seconds, then resumes at the next line of the program |

**Turtle graphics module**

Import the turtle module.

The supported functionality is as follows:

| Function | Description |
|---|---|
| `turtle.forward(steps)` | Moves forward (facing direction) for number of steps |
| `turtle.back(steps)` | Moves backward (opposite-facing direction) for number of steps |
| `turtle.left(degrees)` | Turns anticlockwise the number of degrees |
| `turtle.right(degrees)` | Turns clockwise the number of degrees |
| `turtle.home()` | Moves to canvas origin (0, 0) |
| `turtle.setpos(x, y)` | Positions the turtle at coordinates (x, y) |
| `turtle.reset()` | Clears the drawing canvas, sends the turtle home and resets variables to default values |
| `turtle.hideturtle()` | Makes the turtle invisible |
| `turtle.showturtle()` | Makes the turtle visible |
| `turtle.penup()` | Lifts the pen up |
| `turtle.pendown()` | Puts the pen down |
| `turtle.pensize(width)` | Makes the pen the size of width (positive number) |
| `turtle.pencolor(color)` | Sets the colour of the pen. The input argument can be a string or an RGB colour. For example: "red", "#551A8B", "(0,35,102)" |

## Paper 2 mark scheme

Please note the following regarding the application of this mark scheme:

- Pearson have developed the Programming Language Subset (PLS) document to specify which parts of Python 3 students are required to learn in order to access the assessments. Pearson will not go beyond the scope of the PLS in its assessments. However, if students use alternative valid techniques to answer questions in the assessments, they will also be credited.
- At the time of publication, this qualification uses Python 3 as the underlying programming paradigm for the assessment and teaching and learning of the qualification. Python 2 will not be supported and should not be used in the delivery of this qualification. If at some time in the future a Python 4 is released, we will work closely with our centres before deciding whether to remain with Python 3 or to move to the newer version of the language.
- Google are developing some Google Python library equivalents (gPython) that might be required for Chrome-based schools. Centres that avail of these Google library functions will also be credited.
- The AO breakdown is shown for paper 2 to give greater clarity to the skills demonstrated in each marking point. This is necessary for this practical paper as the majority of the marking points are interlinked in a way that is more complex than in a traditional written paper.

| Question number | Answer | Additional guidance | Mark |
|---|---|---|---|
| **1** | The following assessment objectives are assessed:<br>• AO2.1b<br>• AO3.2b<br><br>Award marks as shown.<br><br>• import random (1)<br>• roll = (1) <integer> 0 (1)<br>• SIDES = (1) <integer> 6 (1)<br>• roll = random.randint(1, SIDES) (1)<br>• print string (1) and roll (1) | • Output of print statements may appear on separate lines<br>• Items in '< >' do not form part of the response, but provide clarification | **(7)** |

```
# ------------------------------------------------------------------
# Import libraries
# ------------------------------------------------------------------

# ===> Complete this line to import the random library
import random

# ------------------------------------------------------------------
# Global variables
# ------------------------------------------------------------------

# ===> Create an integer variable named roll and set it to 0
roll = 0

# ===> Create a constant named SIDES and set it to 6
SIDES = 6

# ------------------------------------------------------------------
# Main program
# ------------------------------------------------------------------

# ===> Assign the result of this library call to the variable roll
roll = random.randint(1, SIDES)

# ===> Display the message "Your roll is" and the variable roll
print ("Your roll is", roll)

```

| Question number | Answer | Additional guidance | Mark |
|---|---|---|---|
| **2** | The following assessment objectives are assessed:<br>• AO2.1a<br>• AO2.1b<br>• AO3.1<br>• AO3.2a<br>• AO3.2b<br>• AO3.2c<br><br>Award marks as shown.<br><br>• Add ':' at end of the line:<br>`if (choice == 'y'):` (1)<br>• Add missing ')' before ':' in the line:<br>`for num in range(5, -1, -1):` (1)<br>• Add missing " before end bracket in the line:<br>`print("Goodbye")` (1)<br>• Printing a suitable question for the user based on context, i.e. "Do you want me to sing?" (1)<br>• Accept user input of 'y' and 'n' (1)<br>• Changing the variable name 'x' to a more meaningful name (1) such as 'choice' throughout the code<br>• Addition of comment indicating reverse stepping (1)<br>• One mark each for insertion of white space to aid readability, up to a maximum of two marks (2)<br>• Correct output for 'y' (count down 5 to 0 and then Goodbye) and correct output for 'n' (Goodbye) (1) | | **(10)** |

```
# ----------------------------------------------------------------
# Global variables
# ----------------------------------------------------------------

# ===>  Change the identifier x to a more meaningful name
choice = ""

# ----------------------------------------------------------------
# Main program
# ----------------------------------------------------------------
# ===> Display a suitable question to the user
print ("Would you like me to sing?")

# ===> Accept the user's input (no validation required)
choice = input("Choose 'y' for yes and 'n' for no")

if (choice == 'y'):
    # ===> Add a comment to explain the effect of the last -1 in this call
    # Counting backwards by using step as -1
    for num in range(5, -1, -1):
        print(num, "green bottles sitting on the wall")

print("Goodbye")

```

| Question number | Answer | Additional guidance | Mark |
|---|---|---|---|
| **3** | The following assessment objectives are assessed:<br>• AO2.1b<br>• AO3.1<br>• AO3.2a<br>• AO3.2b<br>• AO3.2c<br><br>Award marks as shown.<br><br>• Fixing runtime error by coercion of input to 'int' (1)<br>• Fixing errors by using modulus (1)<br>• Use of at least one appropriate 'if' statement in the solution (1)<br>• Adding validation for input numbers using:<br>  ○ relational operator (<=20) (1)<br>  ○ relational operator (>=1) (1)<br>  ○ correct Boolean operator (and/or) (1)<br>• Corrects output message for even numbers and odd numbers (1)<br><br>Levels-based mark scheme to a maximum of 6, from:<br>• Solution design (3)<br>• Functionality (3) | • Fixing error with odd numbers can be done in several different ways (see examples)<br>• Award any accurate tests for validation range<br><br>Considerations:<br>• 6.1.6 Using test data to evaluate a program, such as extreme data [a character], normal data [1...20] and boundary data [0, 21]<br>• 6.2.2 Appropriate use of sequencing, selection and repetition<br>• 6.1.1 Use analysis to solve problems<br>• 6.1.6 Use logical reasoning to evaluate efficiency (i.e. reduce tests) | **(13)** |

**Solution design (levels-based mark scheme)**

| 0 | 1 | 2 | 3 | Max. |
|---|---|---|---|---|
| *No rewardable material* | • There has been little attempt to decompose the problem.<br>• Some of the component parts of the problem can be seen in the solution, although this will not be complete.<br>• Some parts of the logic are clear and appropriate to the problem.<br>• The use of variables and data structures, appropriate to the problem, is limited.<br>• The choice of programming constructs, appropriate to the problem, is limited. | • There has been some attempt to decompose the problem.<br>• Most of the component parts of the problem can be seen in the solution.<br>• Most parts of the logic are clear and appropriate to the problem.<br>• The use of variables and data structures is mostly appropriate.<br>• The choice of programming constructs is mostly appropriate to the problem. | • The problem has been decomposed clearly into component parts.<br>• The component parts of the problem can be seen clearly in the solution.<br>• The logic is clear and appropriate to the problem.<br>• The choice of variables and data structures is appropriate to the problem.<br>• The choice of programming constructs is accurate and appropriate to the problem. | **3** |

**Functionality (levels-based mark scheme)**

| 0 | 1 | 2 | 3 | Max. |
|---|---|---|---|---|
| *No rewardable material* | **Functionality (when the code is run)**<br>• The component parts of the program are incorrect or incomplete, providing a program of limited functionality that meets some of the given requirements.<br>• Program outputs are of limited accuracy and/or provide limited information.<br>• Program responds predictably to some of the anticipated input.<br>• Solution is not robust and may crash on anticipated or provided input. | **Functionality (when the code is run)**<br>• The component parts of the program are complete, providing a functional program that meets most of the stated requirements.<br>• Program outputs are mostly accurate and informative.<br>• Program responds predictably to most of the anticipated input.<br>• Solution may not be robust within the constraints of the problem. | **Functionality (when the code is run)**<br>• The component parts of the program are complete, providing a functional program that fully meets the given requirements.<br>• Program outputs are accurate, informative, and suitable for the user.<br>• Program responds predictably to anticipated input.<br>• Solution is robust within the constraints of the problem. | **3** |

```
# ------------------------------------------------------------------------
# Global variables
# ------------------------------------------------------------------------
num = 0

# ------------------------------------------------------------------------
# Main program
# ------------------------------------------------------------------------

# ---------- Solution 1 ----------
# Coerce the input from string to integer
num = int (input ("Please enter a number (1 ... 20)"))

# Check for valid input numbers
if (num >=1 and num <= 20):
    if (num % 2 == 0):
        print (num, "is even")
    else:
        # If num is not even, then it must be odd
        print (num, "is odd")
else:
    print ("Invalid input")     # Error message for bad input

```

```
# ---------- Solution 2 ----------
num = int (input ("Please enter a number (1 ... 20)"))

if (num < 1):
    print ("Bad input")
elif (num > 20):
    print ("Bad input")
elif (num % 2 == 0):
    print (num, "is even")
else:
    print (num, "is odd")

# ---------- Solution 3 ----------
num = int (input ("Please enter a number (1 ... 20)"))

if (num >= 1) and (num <= 20):
    if (num % 2 == 0):
        print (num, " is even")
    elif (num %2 != 0):
        # Extra check for odd
        print (num, " is odd")
else:
    print ("Bad input")      # Error message for the user

# ---------- Solution 4 ----------
num = int (input ("Please enter a number (1 ... 20)"))

# Keeps looping until a good input is identified
while (num < 1) or (num > 20):
    print ("Invalid input")
    num = int (input ("Please enter a number (1 ...20)"))

if (num % 2 == 0):
    print (num, "is even")
else:
    print (num, "is odd")

```

| Question number | Answer | Additional guidance | Mark |
|---|---|---|---|
| **4** | The following assessment objectives are assessed:<br>• AO2.1b<br>• AO3.1<br>• AO3.2a<br>• AO3.2b<br>• AO3.2c<br><br>Award marks as shown.<br><br>• Use of comments, white space and layout to aid readability (1)<br>• Initial input done outside loop, to handle first entry is '0' (1)<br>• Repetition (while) used as outermost loop (1)<br>• 'elif (year > 13)' is placed later in the logic than 'if (year < 1)' (1)<br>• 'elif (year < 12)' is placed later in the logic than 'elif (year < 7)' (1)<br>• Accepting next round of input done inside loop (1)<br>• Validation messages match validation tests:<br>  ○ Year too small (1)<br>  ○ Year too big (1)<br>• Institution messages match tests:<br>  ○ Primary (1)<br>  ○ Secondary (1)<br>  ○ College (1)<br>• Correct outputs for each set of test data:<br>  ○ 0 = exiting (1)<br>  ○ 1 and 6 = Primary (1)<br>  ○ 7 and 11 = Secondary (1)<br>  ○ 12 = College (1) | | **(15)** |

```
# ------------------------------------------------------------------
# Global variables
# ------------------------------------------------------------------
year = 0            # Do not move this line
strYear = ""        # Do not move this line

# ------------------------------------------------------------------
# Main program
# ------------------------------------------------------------------

# Put the lines into the correct order to solve the problem.
# A user types in a year group.  The program indicates which stage
# of education the year group belongs to.  The program loops until
# the user enters 0.
# Example:
# Input                 Output
# ---------------------------------------
# 0                     Exits program
# 1, 2, 3, 4, 5, 6      Primary
# 7, 8, 9, 10, 11       Secondary
# 12, 13                College

# ---------- Solution 1 ----------
# Prime the loop, just in case the first entry is '0'
strYear = input ("Enter year group (1 to 13, 0 to exit)")
year = int (strYear)

# Keep looping until user wants to stop
while (year != 0):
    # Validate input as a real year group
    if (year < 1):
        print ("Year too small")
    elif (year > 13):
        print ("Year too big")
    elif (year < 7):
        print ("Primary")
    elif (year < 12):
        print ("Secondary")
    else:
        print ("College")

    # Get a new input before going to top of loop
    strYear = input("Enter year group (1 to 13, 0 to exit)")
    year = int(strYear)

```

```
# ---------- Solution 2 ----------
# Prime the loop, just in case the first entry is '0'
strYear = input ("Enter year group (1 to 13, 0 to exit)")
year = int (strYear)

# Keep looping until user wants to stop
while (year != 0):
    # Validate input as a real year group
    if (year < 1):
        print ("Year too small")
    elif (year < 7):
        print ("Primary")
    elif (year < 12):
        print ("Secondary")
    elif (year > 13):
        print("Year too big")
    else:
        print ("College")

    # Get a new input before going to top of loop
    strYear = input("Enter year group (1 to 13, 0 to exit)")
    year = int(strYear)

```

| Question number | Answer | Additional guidance | Mark |
|---|---|---|---|
| **5** | The following assessment objectives are assessed:<br>• AO2.1b<br>• AO3.1<br>• AO3.2a<br>• AO3.2b<br>• AO3.2c<br><br>Award marks as shown.<br><br>• Import of math library (1)<br>• Two parameters in first line of subprogram definition (1) with names 'pRadius' and 'pHeight', in any order (1)<br>• Accurate translation of the formula to code (1)<br>• Use of math.pi constant in formula translation (1)<br>• Two passed-in parameters ('pRadius' and 'pHeight') used in the calculation (1)<br>• Assignment of calculation to 'theVolume' (1)<br>• One return statement with 'theVolume' in brackets (1)<br>• Parameters in call to subprogram are 'baseRadius' and 'coneHeight', in any order (1)<br>• Order of parameters matches order in first line of subprogram definition (1)<br>• Capture of returned value in main program, in 'coneVolume' (1)<br>• Format volume to three decimal places for outputting only (1)<br><br>Levels-based mark scheme to a maximum of 3, from:<br>• Functionality (3) | Considerations:<br>• 6.1.1 Be able to use decomposition to analyse requirements<br>• 6.1.2 Be able to write in a high-level language<br>• 6.6.1 Be able to perform generalisations<br>• Default printing will drop trailing 0s, even if rounded, so string formatting should be used | **(15)** |

**Functionality (levels-based mark scheme)**

| 0 | 1 | 2 | 3 | Max. |
|---|---|---|---|---|
| *No rewardable material* | **Functionality (when the code is run)**<br>• The component parts of the program are incorrect or incomplete, providing a program of limited functionality that meets some of the given requirements.<br>• Program outputs are of limited accuracy and/or provide limited information.<br>• Program responds predictably to some of the anticipated input.<br>• Solution is not robust and may crash on anticipated or provided input. | **Functionality (when the code is run)**<br>• The component parts of the program are complete, providing a functional program that meets most of the stated requirements.<br>• Program outputs are mostly accurate and informative.<br>• Program responds predictably to most of the anticipated input.<br>• Solution may not be robust within the constraints of the problem. | **Functionality (when the code is run)**<br>• The component parts of the program are complete, providing a functional program that fully meets the given requirements.<br>• Program outputs are accurate, informative, and suitable for the user.<br>• Program responds predictably to anticipated input.<br>• Solution is robust within the constraints of the problem. | **3** |

```
# ------------------------------------------------------------------
# Import libraries
# ------------------------------------------------------------------
# ===> Import a library to use Pi
import math

# ------------------------------------------------------------------
# Global variables
# ------------------------------------------------------------------

# Hard coded for testing
coneHeight = 10.7
baseRadius = 1.2
coneVolume = 0.0

# ------------------------------------------------------------------
# Subprograms
# ------------------------------------------------------------------
# ===> Add parameters inside the brackets
def calcVolume (pRadius, pHeight):

    print ("The radius is:", pRadius)
    print ("The height is:", pHeight)

    # ===> Complete the calculation for the volume
    theVolume = 1/3 * math.pi * math.pow (pRadius, 2) * pHeight
    # theVolume = 1/3 * math.pi * pRadius**2 * pHeight
    # theVolume = 1/3 * math.pi * pRadius * pRadius * pHeight

    print ("The volume is:", theVolume)

    # ===> Return the volume to the caller
    return (theVolume)

# ------------------------------------------------------------------
# Main program
# ------------------------------------------------------------------

# ===> Call the subprogram, passing parameters,
#      and catch the returned value in the correct variable
coneVolume = calcVolume (baseRadius, coneHeight)

# ===> Print the total volume to three decimal places using string.format()
# ===> by completing the pattern inside the { }
print ("{:.3f}".format(coneVolume))

```

| Question number | Answer | Additional guidance | Mark |
|---|---|---|---|
| **6** | The following assessment objectives are assessed:<br>• AO2.1b<br>• AO3.1<br>• AO3.2a<br>• AO3.2b<br>• AO3.2c<br><br>Award marks as shown.<br><br>Points-based mark scheme:<br>**Inputs**<br>• Accepts and responds to user input (1)<br>• Validation with range check using relational operators >=1000, <=9999 (1)<br>**Process**<br>• Use of library subprograms len() (1) to work with any number of users in the list<br>• Use of Boolean (1) to stop loop when found or passed over<br>• Use of 2-dimensional indexing (1) in user list<br>**Outputs**<br>• Display of appropriate messages (1)<br><br>Levels-based mark scheme to a maximum of 9, from:<br>• Solution design (3)<br>• Good programming practices (3)<br>• Functionality (3) | Considerations:<br>• 6.1.1 Use decomposition and abstraction to analyse a problem (inputs, outputs, processing, initialisation, design)<br>• 6.6.1 Decompose into subproblems<br>• 6.1.2 Write in a high-level language<br>• 6.2.2 Use sequencing and selection components | **(15)** |

**Solution design (levels-based mark scheme)**

| 0 | 1 | 2 | 3 | Max. |
|---|---|---|---|---|
| *No rewardable material* | • There has been little attempt to decompose the problem.<br>• Some of the component parts of the problem can be seen in the solution, although this will not be complete.<br>• Some parts of the logic are clear and appropriate to the problem.<br>• The use of variables and data structures, appropriate to the problem, is limited.<br>• The choice of programming constructs, appropriate to the problem, is limited. | • There has been some attempt to decompose the problem.<br>• Most of the component parts of the problem can be seen in the solution.<br>• Most parts of the logic are clear and appropriate to the problem.<br>• The use of variables and data structures is mostly appropriate.<br>• The choice of programming constructs is mostly appropriate to the problem. | • The problem has been decomposed clearly into component parts.<br>• The component parts of the problem can be seen clearly in the solution.<br>• The logic is clear and appropriate to the problem.<br>• The choice of variables and data structures is appropriate to the problem.<br>• The choice of programming constructs is accurate and appropriate to the problem. | **3** |

**Good programming practices (levels-based mark scheme)**

| 0 | 1 | 2 | 3 | Max. |
|---|---|---|---|---|
| *No rewardable material* | • There has been little attempt to lay out the code into identifiable sections to aid readability.<br>• Some use of meaningful variable names.<br>• Limited or excessive commenting.<br>• Parts of the code are clear, with limited use of appropriate spacing and indentation. | • There has been some attempt to lay out the code to aid readability, although sections may still be mixed.<br>• Uses mostly meaningful variable names.<br>• Some use of appropriate commenting, although may be excessive.<br>• Code is mostly clear, with some use of appropriate white space to aid readability. | • Layout of code is effective in separating sections, e.g. putting all variables together, putting all subprograms together as appropriate.<br>• Meaningful variable names and subprogram interfaces are used where appropriate.<br>• Effective commenting is used to explain logic of code blocks.<br>• Code is clear, with good use of white space to aid readability. | **3** |

**Functionality (levels-based mark scheme)**

| 0 | 1 | 2 | 3 | Max. |
|---|---|---|---|---|
| *No rewardable material* | **Functionality (when the code is run)**<br>• The component parts of the program are incorrect or incomplete, providing a program of limited functionality that meets some of the given requirements.<br>• Program outputs are of limited accuracy and/or provide limited information.<br>• Program responds predictably to some of the anticipated input.<br>• Solution is not robust and may crash on anticipated or provided input. | **Functionality (when the code is run)**<br>• The component parts of the program are complete, providing a functional program that meets most of the stated requirements.<br>• Program outputs are mostly accurate and informative.<br>• Program responds predictably to most of the anticipated input.<br>• Solution may not be robust within the constraints of the problem. | **Functionality (when the code is run)**<br>• The component parts of the program are complete, providing a functional program that fully meets the given requirements.<br>• Program outputs are accurate, informative, and suitable for the user.<br>• Program responds predictably to anticipated input.<br>• Solution is robust within the constraints of the problem. | **3** |

```
# ----------------------------------------------------------------------
# Global variables
# ----------------------------------------------------------------------

# User Number, Last Name, First Name, Login Name, Passcode
userList = [[110,"Cashin","Bonnie","Cae110",7005],
            [101,"Cheruit","Madeleine","Che101",1507],
            [103,"Chanel","Coco","Cho103",7333],
            [107,"Gres","Madame","Gre107",3054],
            [114,"Hamnett","Katharine","Hae114",4807],
            [118,"Herrera","Carolina","Hea118",5567],
            [111,"Hulanicki","Barbara","Hua111",5125],
            [116,"Johnson","Betsey","Joy116",8869],
            [104,"Lanvin","Jeanne","Lae104",8580],
            [109,"McCardell","Claire","Mce109",5991],
            [102,"Paquin","Jeanne","Pae102",6495],
            [112,"Quant","Mary","Quy112",9028],
            [113,"Rykiel","Sonia","Rya113",1177],
            [105,"Schiaparelli","Elsa","Sca105",2980],
            [108,"Schlee","Valentina","Sca108",6801],
            [106,"Vionnet","Madeleine","Vie106",9042],
            [117,"Von Furstenberg","Diane","Voe117",2553],
            [119,"Wang","Vera","Waa119",2004],
            [115,"Westwood","Vivienne","Wee115",7806]]

inID = ""                # String
inPass = 0               # Integer
found = False            # Haven't found the record yet
passed = False           # Haven't gone past where it should be
index = 0                # The current record being looked at

```

```
# ------------------------------------------------------------------
# Main program
# ------------------------------------------------------------------

# Get user login name
inID = input ("Enter your user login name, type X to exit.")

# Get user passcode
inPass = int (input ("Enter your four digit passcode"))

# Check if passcode is valid
if (inPass >= 1000 and inPass <= 9999):
    # Look through userList to find matching set
    while (found == False and passed == False and index < len(userList)):
        # If both parts match (authenticated), display welcome message
        if (userList[index][3] == inID and userList[index][4] == inPass):
            found = True
            print ("Welcome", userList[index][2], userList[index][1])
        # Check if have passed over where it should be in the list
        elif (userList[index][3] > inID):
            passed = True               # Stops looping
        else:
            index = index + 1           # Look at next entry
    # If not found or passed, display "Invalid Login Credentials"
    if (found == False):
        print ("Invalid Login Credentials")
else:
    print ("Passcode must be four digits long")

```

For information about Pearson Qualifications, including Pearson Edexcel, BTEC and LCCI qualifications visit qualifications.pearson.com

Edexcel and BTEC are registered trademarks of Pearson Education Limited

Pearson Education Limited. Registered in England and Wales No. 872828
Registered Office: 80 Strand, London WC2R 0RL

VAT Reg No GB 278 537121

© Valery Brozhinsky/Shutterstock

ISBN 978-1-4469-6580-1