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

Specimen 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 ▶

S70459A

1/1/1/

S 7 0 4 5 9 A 0 1 0 8

Pearson

Answer ALL questions.

Suggested time: 15 minutes

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:

$$circumference = 2\pi r$$

- π 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 the round() function.

Do **not** add any additional functionality.

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

(Total for Question 1 = 10 marks)

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? ")
  ```
- amend original line 20 to convert to uppercase
  ```
  initials = initials.lower ()
  ```
- change the variable `a` to a more meaningful name
- add a comment to original line 21, to explain what it does.

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 to print a horizontal histogram showing the frequency of the vowels (A, E, I, O, U) in a string. The user enters the string.

The string will fit on one line only and will not include a carriage return.

For the string 'The quick brown fox jumps over the lazy dog', this histogram shows that 'A' occurs only once and 'E' occurs exactly three times.

The output must be formatted like the histogram. The output will vary according to the user's input.

```
A | A
E | EEE
I | I
O | OOOO
U | UU
```

Open file **Q03.py**

Amend the code to:

- create two one-dimensional data structures, implemented as lists
- complete calls to built-in subprograms
- complete a call to a user-devised subprogram
- use selection to check for each vowel and increment the corresponding count
- ensure the program behaves predictably if the user enters an empty string or a string without any vowels, by displaying the vertical axis only.

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 An International Standard Book Number (ISBN) is a unique identifier for commercially published books. Here is an example of an ISBN barcode.

© PAL

A program has been written to calculate the check digit for a 10-digit ISBN.

The user enters the first nine digits of the ISBN. A check digit is calculated and appended to the original digits. The numeric value of the check digit is calculated by:

- multiply ten times the first digit from the left, nine times the second digit from the left, and onwards until two times the furthest digit from the left
- add all the products together
- apply a modulus 11 to the sum of all the products
- calculate the numeric value of the check digit by subtracting the result of the modulus operation from 11
- when the numeric value of the check digit is 11, the string '0' is appended to the ISBN
- when the numeric value of the check digit is 10, the string 'X' is appended to the ISBN
- in all other cases, the string value of the check digit is appended to the ISBN.

This table shows inputs and required outputs for a fully functional program.

No validation of inputs is required.

Input	Output
071954400	Check digit = 9 ISBN = 0719544009
061826941	Check digit = X ISBN = 061826941X
047119047	Check digit = 0 ISBN = 0471190470

The lines of code in the program are mixed up. The comments are in the right order to match the logic of the solution.

Open file **Q04.py**

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 A set of data is stored in a comma-separated value text file. Each line in the file is made up of ten integers. All numbers are in the range of zero to 100, inclusive. The file has multiple lines.

A program is required to calculate and report the mean of each line.

A program and subprograms have been started to carry out the processing.

This is what the output must look like. Column widths may vary, but alignment and number formats should match the table.

Row	Mean
1	33.9Ø
2	55.3Ø
3	41.5Ø
4	48.1Ø
5	54.5Ø

Open file **Q05.py**

Amend the program and subprograms to meet the following requirements:

- all data in Q05_Data.txt must be processed
- only local variables must be created, i.e. do not create any global variables
- results must be displayed in columnar format, using <string>.format().

Do **not** add any additional functionality.

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

(Total for Question 5 = 12 marks)

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)

TOTAL FOR PAPER = 75 MARKS

DO NOT WRITE IN THIS AREA

DO NOT WRITE IN THIS AREA

DO NOT WRITE IN THIS AREA

BLANK PAGE