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 ▶

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:

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

- V is volume

- π 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