

Paper reference 1CP2/02

Pearson Edexcel

Level 1/Level 2 GCSE (9–1)

Computer Science

PAPER 2: Application of

Computational Thinking

Thursday 25 May 2023 – Afternoon

Time: 2 hours

X72596RA

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 Programming Language Subset (PLS) document (enclosed).

YOU WILL BE GIVEN

Data Book.

Model for Question 4.

INSTRUCTIONS

- Answer ALL questions on your computer.
- Save the new or amended code in the 'COMPLETE CODING' folder using the name given in the question.
- Do NOT overwrite the original code and data files provided to you.
- You must NOT use the internet at any time during the examination.

(continued on the next page)

Turn over

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.
- The ‘STUDENT CODING’ folder in your user area includes all the code and data files you need.

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.
-

ANSWER ALL QUESTIONS.

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%.

(continued on the next page)

Question 1 continued

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 = 7 marks)

SUGGESTED TIME: 15 MINUTES

2. A program is written for an exercise routine. It displays the names of warm – up exercises, stored in an array. The user enters a number. That number of exercises is selected randomly from the array and displayed.

Open file **Q02.py**

Amend the code to:

- fix the syntax error on original line 4
import random
- fix the syntax error on original line 16
for exercise in exerciseTable
- complete original line 20 to generate a random number between 0 and 4
index = random.

(continued on the next page)

Question 2 continued

- complete original line 8 to make the exercise names be string data types
**exerciseTable = ["squats", "planks",
pushups,"lunges",
"burpees"]**
- fix the IndexError on original line 21
name = exerciseTable [index + 1]
- fix the NameError on original line 22
print (naime)
- fix the logic error on original line 19 that causes one less exercise to be printed than is asked for
for count in range (numExercises – 1):
- use white space to improve the readability of the code.

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

Do NOT add any additional functionality.

Save your amended code file as Q02FINISHED.py

(Total for Question 2 = 8 marks)

SUGGESTED TIME: 20 MINUTES

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.

(continued on the next page)

Question 3 continued

Open file Q03.py

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

You will need to:

- **amend some lines of code**
- **choose between alternative lines of code.**
Make a choice by removing the # at the beginning of the line you choose to execute
- **run the program at least twice and check the output file each time to make sure it meets the requirements.**

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

Do NOT add any additional functionality.

Save your amended code as Q03FINISHED.py

(Total for Question 3 = 15 marks)

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

Ask for the model for Question 4.

Look at the diagrams for Question 4 in the separate Data Book.

Diagram 1 shows a triangular prism,

Diagram 2 shows the cross – section and

Diagram 3 shows the side view.

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

(continued on the next page)

Question 4 continued

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

THE PROGRAM MUST MEET THESE REQUIREMENTS:

- take three decimal inputs from the user
 - all inputs must be greater than zero
- check for invalid inputs, using relational and logical operators
- display an error message if an input is invalid. Invalid input should not be processed
- process all valid inputs
- calculate the area of the triangle
- display the area of the triangle, rounded to two decimal places

(continued on the next page)

Turn over

Question 4 continued

- calculate the volume of the prism
- display the volume of the prism using the `<string>.format()` function in eight columns with two decimal places. Include the words 'cubic units' after the volume
- in all cases, display a goodbye message just before terminating.

Test the functionality of the program using the data in this table.

	b	h	l	A	V
PRISM 1	4.567	1.23	89.01	2.81	250.00
PRISM 2	2.74	6.01	5.55	8.23	45.70

(continued on the next page)

Question 4 continued

Open file Q04.py

Amend the code to meet the requirements.

Do NOT add any additional functionality.

Save your amended code as Q04FINISHED.py

(Total for Question 4 = 15 marks)

SUGGESTED TIME: 25 MINUTES

5. A program is being developed to generate a user identification string.

The letter part of the identification string is made up of the last name joined with the first letter of the first name. All letters must be in lowercase.

The number part of the identification string is the sum of the ASCII values for each of the digits in the date of birth (ddmmyyyy).

The identification string for the user Viola Bassir, born 15th June 2005, is bassirv403, all in lowercase.

Open file **Q05.py**

Amend the code to:

- Ensure local and global variables with the same names are not confused
 - change the names of the local variables to distinguish them from the global variables with the same name

(continued on the next page)

Turn over

Question 5 continued

- **Welcome the user**
 - add a procedure, with no parameters, to display a welcome message for the user
 - call the welcome procedure before taking input from the user
- **Ensure the last name and first name are all lowercase**
 - convert last name and first name to lowercase after they are inputted by the user
- **Validate the date of birth in the main program using the built-in string manipulation subprograms**
 - check that only the digits 0 to 9 appear in the date of birth
 - call the `makeID ()` function if the date of birth is valid
 - tell the user if the date of birth is invalid. Invalid input should not be processed

(continued on the next page)

Question 5 continued

- **Generate the correct number part of the identification string in the `makeID()` function**
 - **correct the logic error caused by using the `int()` function in the number part calculation rather than using a function that returns the ASCII value of the character**

Do NOT add any additional functionality.

Save your amended code file as `Q05FINISHED.py`

(Total for Question 5 = 15 marks)

SUGGESTED TIME: 25 MINUTES

6. A program is required to determine if a user can access a database. The names and passwords of users are stored in a two – dimensional array.

Open file Q06.py

Write a program to meet these requirements.

INPUTS

- Prompt for and accept a user name and a password
 - neither should be blank

PROCESS

- Implement authentication by searching the array for the user's name and password
 - ensure the search works for any length of array

(continued on the next page)

Question 6 continued

OUTPUT

- **Display a suitable message when the correct combination of name and password is found**
- **Display a suitable message when the user's name is found but the password does not match**
- **Display a suitable message when the user's name is not found**

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

(Total for Question 6 = 15 marks)

TOTAL FOR PAPER = 75 MARKS

END OF PAPER
