Please write clearly, in BLOCK CAPITALS and black ink

Centre number ________ **Candidate number** ____________________

First name (s) ____

Surname ____

Date of Exam _______________ **Time allowed: 2 hours**

GCSE (9–1) Computer Science

1CP2/02 Paper 2: Application of Computational Thinking

Total Marks

PAPER 2A

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 ( )
- The student version of this paper has **8** pages

ADVICE

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

Answer ALL questions.

Make sure you have a copy of the Programming Language Subset (PLS) document before you begin these questions.

Suggested time 10 minutes

1 A program calculates the area of a circle with radius 50 using the formula:

πr^2

The program uses the constant pi to calculate the area of the circle.

Open file **Q01**.

Amend the code to add or complete lines to:

- Import the math module
- Create a constant and assign the math module's value for π to it
- Create one variable with the initial value 50
- Calculate the area of the circle
- Display the area of the circle on the screen

Do **not** add any additional functionality.

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

(Total for Question 1 = 8 marks)

Suggested time 20 minutes

2 A program asks the user their height in metres. It then asks if they would like to display their height in centimetres or inches. If the user enters 'cm' then the height is displayed in centimetres. If they enter 'inch' then the height is displayed in inches.

Open file **Q02**.

Amend the code to add or complete lines to:

- Fix the syntax error in line 15
- Fix the syntax error in line 32
- Change the identifier x to a more meaningful name
- Convert the user's input to a floating point number
- Display a suitable question to the user for their choice of 'cm' or 'inch'
- Accept the user's input
- Replace 2.54 with a constant
- Make two uses of white space to aid readability
- If the user enters 'cm' convert the number of metres they entered to centimetres
- Output the result and word 'centimetres'

Do **not** add any additional functionality.

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

(Total for Question 2 = 11 marks)

Suggested time 20 minutes

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

- Ask the user to enter a new PIN (personal identification number)
- Only accept numbers that are exactly four digits long
- Print ‘invalid input’ if the output is not the correct length. You can assume that only numbers will be entered
- Print ‘the PIN cannot have all four numbers the same’ if the PIN entered has four numbers that are the same – for example, 3333 would not be allowed
- Print ‘New PIN accepted’ if the number is four digits long and does not have four numbers that are the same

Open file **Q03**.

Amend the code to add or complete lines to:

- Fix the runtime error caused by inputting a PIN such as 0023
- Validate that a PIN has four digits
- Fix the logic error that checks that all four numbers are the same
- Output ‘New PIN accepted’ if the PIN is valid
- Make sure all output messages are the same as above
- Add one comment to explain how the code works to check the PIN doesn’t contain four numbers that are all the same

Do **not** add any additional functionality.

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

(Total for Question 3 = 14 marks)

Suggested time 20 minutes

4 A program asks the user to input the first three letters of the month. If the month is ‘feb’ the program also asks the user to input if it is a leap year. The program then outputs the number of days in the month.

The program loops continually until a valid month name is entered.

Input	Output
jan, mar, may, jul, aug, oct, dec	31 days
apr, jun, sep, nov	30 days
Month: feb Leap year: y	29 days
Month: feb Leap year: n	28 days

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 = 13 marks)

Suggested time 25 minutes

5 A program calculates the average of a list of monthly sales.

The 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 list of monthly sales passed as a parameter. You can assume that a list passed to the subprogram will always contain numbers, however, the list can be of any length. No validation is required.
- The subprogram must calculate the average of all the sales given as a parameter.
- The main program must print the average of the sales, formatted to show two decimal places with a £ sign at the start (e.g. £26012.21).

Do **not** add any additional functionality.

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

(Total for Question 5 = 14 marks)

Suggested time 25 minutes

6 A program is needed to search for records for a last name given by the user.

The program stores a set of names and telephone numbers in a comma separated value (CSV) text file.

Each record in the CSV file stores the contact’s first name, last name and phone number. There are 17 records in the file.

The records are not sorted.

Open file **Q06**.

Write a program to meet the following requirements:

Inputs

- Prompt for and accept a last name, no validation required
- Open the file “phonenumbers.csv” in read only mode – this has already been carried out for you in the file **Q06**

Process

- Work with any number of records in the list
- Use a linear search to find any records that match the last name entered by the user
- Continue to search through the list to find all records that match the last name
- The program needs to find a match on any variation of capitals that are used in the file or search name. For example, if “suLLivan” is entered as the search term, this should match the record “Sullivan” in the file

Outputs

- If a matching record is found, display the fields for the record. For example, if a search is made for “Rankin”, the following should be output:

```
Enter the last name: Rankin
First name: Stuart
Last name: Rankin
Phone number: 01214960679
```

- If more than one record is found, each one should be displayed. For example:

```
Enter the last name: Palmer
First name: Ailsa
Last name: Palmer
Phone number: 01184960997

First name: Roscoe
Last name: Palmer
Phone number: 01914980647
```

- If a match is found, display the number of records matched as follows:
  Name found in 2 records.
- If the name is not found, inform the user as follows:
  Name not found.
- Inform the user how many records were searched:
  17 records searched.

Do **not** add any additional functionality.

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

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

(Total for Question 6 = 15 marks)

TOTAL FOR PAPER = 75 MARKS