Pleasechecktheexaminationdetailsbelowbeforeenteringyourcandidateinformation

Candidatesurname	Othernames

Pearson Edexcel International GCSE (9–1)

CentreNumber | CandidateNumber

3–5 June 2020

Time: 3 hours | Paper Reference **4CP0/02**

Computer Science

Paper 2: Application of Computational Thinking

You must have:
A computer workstation with appropriate programming language code editing software and tools, including interpreter/compiler, CODES folder containing code and data files, pseudocode command set (enclosed)

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 **requiring a written answer** in the spaces provided – *there may be more space than you need.*
- Only **one** programming language (Python, C# and Java) must be used throughout the test.
- Carry out practical tasks on the computer system and save new or amended code using the name given in the question with the appropriate file extension.
- Do **not** overwrite the original code and data files provided to you.
- You must **not** use the internet during the test.

Information

- The total mark for this paper is 80.
- The marks for **each** question are shown in brackets – *use this as a guide as to how much time to spend on each question.*
- This paper covers Python, C# and Java.
- The CODES folder in your user area includes all the code and data files you need.
- The invigilator will tell you where to store your work.

Advice

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

Turn over ▶

P61885A

1/1/1/1/1

P 6 1 8 8 5 A 0 1 1 6

Pearson

Answer all questions.

Answer the questions **requiring a written answer** 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 ☒.

Carry out practical tasks on the computer system and save new or amended code using the name given with the appropriate file extension.

Use only ONE programming language throughout the examination.

Indicate the programming language that you are using with a cross in a box ☒.

C#	☒	Java	☒	Python	☒

1 Computer programs make use of many programming constructs.

(a) Identify which **one** of these is **not** an arithmetic operator.

(1)

☒ **A** add

☒ **B** divide

☒ **C** equals

☒ **D** modulus

(b) State the values used by the Boolean data type.

(1)

.....

.....

(c) Open **Q01c** in the code editor.

The program should multiply two integers.

There are **three** errors in the code.

Amend the code to correct the errors.

Save your amended code as **Q01cFINISHED** with the correct file extension for the programming language.

(3)

(d) State what is meant by the term 'variable'.

(1)

.....

.....

(e) Ayesha has written a program that calculates the sum of a set of numbers.

Open **Q01e** in the code editor.

Answer these questions about the code.

(i) Identify the line number where the code includes a logic operator.

(1)

.....

(ii) Identify the line number where a subprogram definition starts.

(1)

.....

(iii) Identify the name of a numeric variable.

(1)

.....

.....

(iv) Identify the name of a string variable.

(1)

.....

.....

(Total for Question 1 = 10 marks)

2 Rama wants some computer programs for his daughter, Ayomi.

(a) Rama wants a program to calculate the area of a triangle.

(area of a triangle = ½ base x height)

This pseudocode contains the logic required to create the program.

```
 1 #  Initialise variables
 2
 3 SET base TO 50
 4 SET heightChk TO TRUE
 5
 6 #  Print prompt and take input from user
 7
 8 WHILE heightChk DO
 9     SEND "Enter the height (between 1 and 100): " TO DISPLAY
10     RECEIVE height FROM (INTEGER) KEYBOARD
11     IF (height >= 1 AND height <= 100)
12     THEN heightChk = FALSE
13     END IF
14 END WHILE
15
16 #  Calculate and print out values
17
18 SET area TO 0.5 * base * height
19
20 SEND ("Base of triangle is: " , base) TO DISPLAY
21 SEND ("Height of triangle is: " , height) TO DISPLAY
22 SEND ("Area of triangle is: " , area) TO DISPLAY
```

Write a program to implement the logic in the pseudocode.

Open **Q02a** in the code editor.

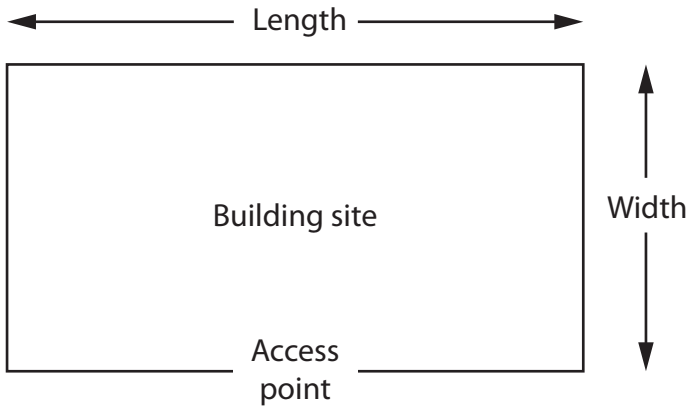
You must use the structure given in **Q02a** to write the program.

Do not add any further functionality.

Save your code as **Q02aFINISHED** with the correct file extension for the programming language.

(10)

(b) Ayomi wants a program that will calculate the number of fence panels required to surround building sites such as the one shown in the diagram (not to scale).



Each building site is a rectangle.

Each fence panel is 1 metre long.

A gap of 4 metres is needed to access each site. The location of the gap is not significant.

The user will input the length and width of the building site as whole numbers of metres.

The program will output the minimum number of full panels needed.

Q02b provides a structure for the program.

Open **Q02b** in the code editor.

Write the program.

You must use the structure given in **Q02b** to write the program.

Do not add any further functionality.

Save your code as **Q02bFINISHED** with the correct file extension for the programming language.

(7)

(Total for Question 2 = 17 marks)

3 Several encryption algorithms have been developed.

(a) State what is meant by the term 'encryption'.

(1)

.....

.....

(b) Give **one** reason why data may need to be encrypted.

(1)

.....

.....

(c) (i) Perform a Rail Fence cipher encryption of the text **COMPUTATIONAL** using a key of 4.

Show your working.

(4)

Encrypted text:

.....

(ii) Explain **one** reason why the Rail Fence cipher is a weak encryption algorithm.

(2)

.....

.....

.....

.....

(Total for Question 3 = 8 marks)

4 Gianluca is a schoolteacher. He is in charge of a year group of pupils.

(a) He has started to create a program to analyse test results.

Open **Q04a** in the code editor.

Amend the code to complete the 'if statement' used to produce the outputs described in this table.

Condition	Output text
English test score was less than 40; Maths test score was less than 50	Student failed both tests
English test score was less than 40 or Maths test score was less than 50	Student failed one test
English test score was at least 80; Maths test score was at least 85	Student passed both tests with distinction
Other conditions	Student passed both tests

Do not add any further functionality.

Save your code as **Q04aFINISHED** with the correct file extension for the programming language.

(6)

(b) Gianluca has started to create a program to analyse pupil attendance.

He wants to use a menu system with subprograms.

Open **Q04b** in the code editor.

The program must include subprograms to:

- display the **names** of students whose percentage attendance was less than 75%
- count and display the **number** of students whose percentage attendance was 90% or higher.

Complete the program code to implement these requirements using the structure given in **Q04b**.

Do not add any further functionality.

Save your code as **Q04bFINISHED** with the correct file extension for the programming language.

(7)

(Total for Question 4 = 13 marks)

5 Cerys Jones is the manager of a shop that sells electrical goods.

She is creating a stock control program.

(a) Each item of stock will have a ten-character code.

Each code consists of three parts.

Part	Description
Characters 1 to 3	Uppercase letters
Characters 4 to 9	Numbers between 1 and 9 inclusive
Character 10	'e' or 'o' indicating whether the sum of the numbers is even or odd

WAS123456o, FDG232324e and VAC129222e are valid codes.

All codes need to be validated.

Complete the table to show **two** additional validation tests.

For each test, give **one** example of erroneous data. The example that you give should fail **only** that test.

(4)

Validation test	Erroneous data
Is the last character 'o' or 'e'?	ABC654321v

(b) Cerys will need an algorithm to search the table of stock data.

(i) State **two** advantages of using a linear search algorithm rather than a binary search algorithm.

(2)

1 .....

.....

2 .....

.....

(ii) State **two** disadvantages of using a linear search algorithm rather than a binary search algorithm.

(2)

1 .....

.....

2 .....

.....

(c) Cerys will use functions and procedures in her program.

(i) One similarity between a function and a procedure is that they are both subprograms.

State **two** other similarities.

(2)

1 .....

.....

2 .....

.....

(ii) Explain **one** difference between a function and a procedure.

(2)

.....

.....

.....

.....

(Total for Question 5 = 12 marks)

BLANK PAGE

QUESTION 6 BEGINS ON THE NEXT PAGE.

6 Farshia is the regional manager for an insurance company.

She manages a team of sales staff.

She wants a program to analyse the performance of her team over a number of months.

Test data has been included in the code.

Open **Q06** in the code editor.

Write a program to:

- calculate and display the total sales made by each member of the team
- calculate and display the total sales made by the whole team
- display the first name, last name and the total sales made by the two members of the team with the highest total sales. (Ignore the possibility of two or more members of the team having the same total sales.)

Your program should function correctly even if the number of months or number of members of the team is changed.

Save your code as **Q06FINISHED** with the correct file extension for the programming language.

(20)

You may use this space for planning / design work.

(Total for Question 6 = 20 marks)

TOTAL FOR PAPER = 80 MARKS

BLANK PAGE

BLANK PAGE

BLANK PAGE

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

3–5 June 2020

Paper Reference **4CP0/02**

Computer Science

Component 2

Pseudocode command set

Resource Booklet

Do not return this resource booklet with the question paper.

Turn over ▶

P61885A

1/1/1/1/1

P 6 1 8 8 5 A

Pearson

Pseudocode command set

Questions in the written examination that involve code will use this pseudocode for clarity and consistency. However, students may answer questions using any valid method.

Data types

INTEGER

REAL

BOOLEAN

CHARACTER

Type coercion

Type coercion is automatic if indicated by context. For example 3 + 8.25 = 11.25 (integer + real = real)

Mixed mode arithmetic is coerced like this:

	INTEGER	REAL
INTEGER	INTEGER	REAL
REAL	REAL	REAL

Coercion can be made explicit. For example, RECEIVE age FROM (INTEGER) KEYBOARD assumes that the input from the keyboard is interpreted as an INTEGER, not a STRING.

Constants

The value of constants can only ever be set once. They are identified by the keyword CONST. Two examples of using a constant are shown.

CONST REAL PI

SET PI TO 3.14159

SET circumference TO radius * PI * 2

Data structures

ARRAY

STRING

Indices start at zero (0) for all data structures.

All data structures have an append operator, indicated by &.

Using & with a STRING and a non-STRING will coerce to STRING. For example, SEND 'Fred' & age TO DISPLAY, will display a single STRING of 'Fred18'.

Identifiers

Identifiers are sequences of letters, digits and '_', starting with a letter, for example:
MyValue, myValue, My_Value, Counter2

Functions

LENGTH()

For data structures consisting of an array or string.

RANDOM(n)

This generates a random number from 0 to n.

Comments

Comments are indicated by the # symbol, followed by any text.

A comment can be on a line by itself or at the end of a line.

Devices

Use of KEYBOARD and DISPLAY are suitable for input and output.

Additional devices may be required, but their function will be obvious from the context.
For example, CARD_READER and MOTOR are two such devices.

Notes

In the following pseudocode, the < > indicates where expressions or values need to be supplied. The < > symbols are not part of the pseudocode.

Variables and arrays		
Syntax	**Explanation of syntax**	**Example**
SET Variable TO <value>	Assigns a value to a variable.	SET Counter TO 0 SET MyString TO 'Hello world'
SET Variable TO <expression>	Computes the value of an expression and assigns to a variable.	SET Sum TO Score + 10 SET Size to LENGTH(Word)
SET Array[index] TO <value>	Assigns a value to an element of a one-dimensional array.	SET ArrayClass[1] TO 'Ann' SET ArrayMarks[3]TO 56
SET Array TO [<value>, …]	Initialises a one-dimensional array with a set of values.	SET ArrayValues TO [1, 2, 3, 4, 5]
SET Array [RowIndex, ColumnIndex] TO <value>	Assigns a value to an element of a two dimensional array.	SET ArrayClassMarks[2,4] TO 92

Selection		
Syntax	**Explanation of syntax**	**Example**

IF <expression> THEN
  <command>
END IF
	If <expression> is true then command is executed.	IF Answer = 10 THEN
		  SET Score TO Score + 1
		END IF

IF <expression> THEN
  <command>
ELSE
  <command>
END IF
	If <expression> is true then first <command> is executed, otherwise second <command> is executed.	IF Answer = 'correct' THEN
		  SEND 'Well done' TO DISPLAY
		ELSE
		  SEND 'Try again' TO DISPLAY
		END IF

Repetition		
Syntax	**Explanation of syntax**	**Example**

WHILE <condition> DO
  <command>
END WHILE
	Pre-conditioned loop. Executes <command> whilst <condition> is true.	WHILE Flag = 0 DO
		  SEND 'All well' TO DISPLAY
		END WHILE

REPEAT <command> UNTIL <expression>	Post-conditioned loop. Executes <command> until <condition> is true. The loop must execute at least once.	REPEAT SET Go TO Go + 1 UNTIL Go = 10
REPEAT <expression> TIMES <command> END REPEAT	Count controlled loop. The number of times <command> is executed is determined by the expression.	REPEAT 100-Number TIMES SEND '*' TO DISPLAY END REPEAT

FOR <id> FROM <expression> TO
<expression> DO
  <command>
END FOR
	Count controlled loop. Executes <command> a fixed number of times.	FOR Index FROM 1 TO 10 DO
		  SEND ArrayNumbers[Index] TO DISPLAY
		END FOR

FOR <id> FROM <expression> TO
<expression> STEP <expression> DO
  <command>
END FOR
	Count controlled loop using a step.	FOR Index FROM 1 TO 500 STEP 25 DO
		  SEND Index TO DISPLAY
		END FOR

FOR EACH <id> FROM <expression> DO <command> END FOREACH	Count controlled loop. Executes for each element of an array.	SET WordsArray TO ['The', 'Sky', 'is', 'grey'] SET Sentence to '' FOR EACH Word FROM WordsUArray DO SET Sentence TO Sentence & Word & ' ' END FOREACH

Input/output		
Syntax	**Explanation of syntax**	**Example**
SEND <expression> TO DISPLAY	Sends output to the screen.	SEND 'Have a good day.' TO DISPLAY
RECEIVE <identifier> FROM (type) <device>	Reads input of specified type.	RECEIVE Name FROM (STRING) KEYBOARD RECEIVE LengthOfJourney FROM (INTEGER) CARD_READER RECEIVE YesNo FROM (CHARACTER) CARD_READER

File handling		
Syntax	**Explanation of syntax**	**Example**
READ <File> <record>	Reads in a record from a <file> and assigns to a <variable>. Each READ statement reads a record from the file.	READ MyFile.doc Record
WRITE <File> <record>	Writes a record to a file. Each WRITE statement writes a record to the file.	WRITE MyFile.doc Answer1, Answer2, 'xyz 01'

Subprograms		
Syntax	**Explanation of syntax**	**Example**
PROCEDURE <id> (<parameter>, …) BEGIN PROCEDURE <command> END PROCEDURE	Defines a procedure.	PROCEDURE CalculateAverage (Mark1, Mark2, Mark3) BEGIN PROCEDURE SET Avg to (Mark1 + Mark2 + Mark3)/3 END PROCEDURE

FUNCTION <id> (<parameter>, …)
BEGIN FUNCTION
  <command>
RETURN <expression>
END FUNCTION
	Defines a function.	FUNCTION AddMarks (Mark1, Mark2, Mark3)
		BEGIN FUNCTION
		  SET Total to (Mark1 + Mark2 + Mark3)/3
		  RETURN Total
		END FUNCTION

<id> (<parameter>, …)	Calls a procedure or a function.	Add (FirstMark, SecondMark)

Arithmetic operators	
Symbol	**Description**
+	Add
-	Subtract
/	Divide
*	Multiply
^	Exponent
MOD	Modulo
DIV	Integer division

Relational operators	
Symbol	**Description**
=	equal to
<>	not equal to
>	greater than
>=	greater than or equal to
<	less than
<=	less than or equal to

Logical operators	
Symbol	**Description**
AND	Returns true if both conditions are true.
OR	Returns true if any of the conditions are true.
NOT	Reverses the outcome of the expression; true becomes false, false becomes true.

BLANK PAGE