Paper 2: Application of Computational Thinking

Question number	Answer	Additional guidance	Mark
1(a)(i)	myNumbers	Candidates are required to open the file Q01a in the code editor.	(1)

Question number	Answer	Additional guidance	Mark
1(a)(ii)	**Python** Lines 3–8/3 **Java** Lines 7–14/7 **C#** Lines 10–17/10	Candidates are required to open the file Q01a in the code editor.	(1)

Question number	Answer	Additional guidance	Mark
1(a)(iii)	**Python** Lines 5–8/5 **Java** Lines 10–14/10 **C#** Lines 13–16/13	Candidates are required to open the file Q01a in the code editor.	(1)

Question number	Answer	Additional guidance	Mark
1(a)(iv)	myNumbers/i/total	Candidates are required to open the file Q01a in the code editor.	(1)

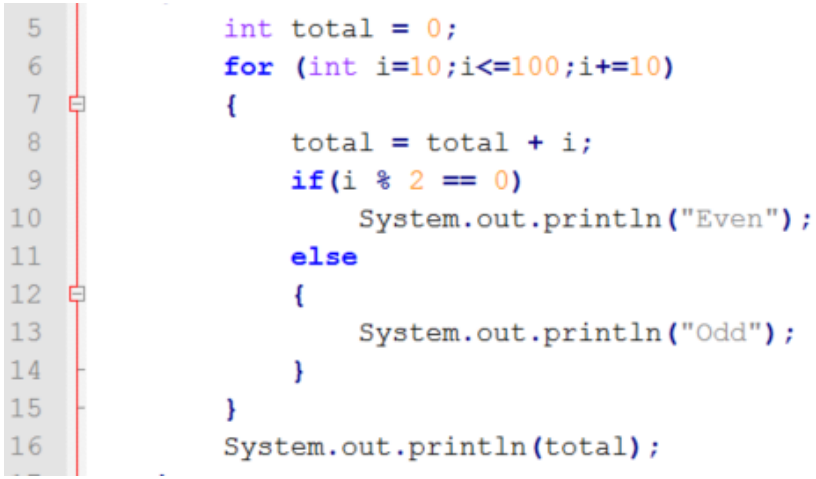
Question number	Answer	Additional guidance	Mark
1(b)(i)	Any **one** from: • comments/annotations • descriptive variable names.	Candidates are required to open the file Q01b in the code editor.	**(1)**

Question number	Answer	Additional guidance	Mark
1(b)(ii)	Any **one** from: • another programmer could understand it • future maintenance would be easier • easier for another programmer to fix bugs/make amendments • less likely to introduce bugs yourself.		**(1)**

Question number	Answer	Additional guidance	Mark
1(c)(i)	Any **one** from: • an error in following the rules of the programming language • not following the grammar rules of the programming language • not being able to translate a line of code because of an error in using the language • misspelling command words in the programming language.		**(1)**

Question number	Answer	Additional guidance	Mark

1(c)(ii)	**Python**
	• Total is undefined/need to add initialisation for variable total (1).
	• Equals symbol in If statement needs to be replaced with '==' (1).
	• Print ("Odd") needs to be indented (1).
	
	`myNumbers = [10, 20, 30, 40 ,50, 60, 70, 80, 90, 100]`
	`total = 0`
	`for theNumber in myNumbers:`
	`    total = total + theNumber`
	`    if(theNumber % 2 == 0):`
	`        print("Even")`
	`    else:`
	`        print("Odd")`
	`print(total)`
		Candidates are required to open the file Q01c in the code editor. Amended code should be saved as Q01cFINISHED.
		
		Do not penalise logic errors such as initialising total inside loop.
			(3)

	Java • Total is undefined/need to add initialisation for variable total (1). • Equals symbol in If statement needs to be replaced with '==' (1). • Missing closed } after Print ("Odd") (1). 		

```
5           int total = 0;
6           for (int i=10;i<=100;i+=10)
7           {
8               total = total + i;
9               if(i % 2 == 0)
10                  System.out.println("Even");
11              else
12              {
13                  System.out.println("Odd");
14              }
15          }
16          System.out.println(total);
```

	C#
	• Total is undefined/need to add initialisation for variable total (1).
	• Equals symbol in If statement needs to be replaced with '==' (1).
	• Missing End If after Console.WriteLine ("Odd") (1).
	
	`int total = 0;`
	`for (int i = 10; i <= 100; i += 10)`
	`{`
	`  total = total + i;`
	`  if (i % 2 == 0)`
	`  Console.WriteLine("Even");`
	`  else`
	`  {`
	`  Console.WriteLine("Odd");`
	`  }`
	`}`
	`Console.WriteLine(total);`
	`Console.ReadKey();`		

Question number	Answer	Additional guidance	Mark

2(a)	Award 1 mark for each of:
	
	• attempting to input country and print country (1)
	• printing string plus country (1)
	• attempting to input number of children and number of adults (1)
	• coercion of at least one data type (1)
	• attempting to calculate and print a total (1)
	• calculating a total using the addition operator (1)
	• using two variables (1)
	• printing a string plus an integer (1)
	• compiling without syntax errors (1)
	• executing and producing the correct output (1).
	
	Python
	
	`# Print prompt and take country from user`
	`country = input ("Enter the country you're visiting from: ")`
	
	`# Tell the user their country`
	`print ("You are from: ", country)`
	
	`# Take number of adults in party from user`
	`adult = int (input ("Enter the number of adults in your party: "))`
	
	`# Take number of children in party from user`
	`children = int (input ("Enter the number of children in your party: "))`
	
	`# Calculate total number in party`
	`total = adult + children`
	
	`# Tell the user the total number of people in their party`
	`print ("The total in your party is: ", total)`
		Candidates are required to open the file Q02a in the code editor. Amended code should be saved as Q02aFINISHED.
		
		Logic of algorithm must be followed as set out. Alternatives must address each point.
		
		Do not penalise candidates who attempt more than the stated requirements.
			(10)

Java

```
6           Scanner input = new Scanner(System.in);
7
8           // Print prompt and take country from user
9           System.out.print("Enter the country you are visiting from: ");
10          String country = input.next();
11
12          // Tell the user their country
13          System.out.println("You are from: " + country);
14
15          // Take number of adults in party from user
16          System.out.print("Enter the number of adults in your party: ");
17          int adults = input.nextInt();
18
19          // Take number of children in party from user
20          System.out.print("Enter the number of children in your party: ");
21          int children = input.nextInt();
22
23          // Calculate total number in party
24          int total = adults + children;
25
26          // Tell the user the total number of people in their party
27          System.out.println("The total in your party is: " + total);
28
```

C#

```
8           // Print prompt and take country from user
9           Console.WriteLine("Enter the country you are visiting from: ");
10          String country = Console.ReadLine();
11          Console.WriteLine("You are from: " + country);
12
13          // Tell the user their country
14          Console.WriteLine("Enter the number of adults in your party: ");
15
16          // Take number of adults in party from user
17          int adults = Convert.ToInt32(Console.ReadLine());
18          Console.WriteLine("Enter the number of children in your party: ");
19
20          // Take number of children in party from user
21          int children = Convert.ToInt32(Console.ReadLine());
22
23          // Calculate total number in party
24          int total = adults + children;
25
26          // Tell the user the total number of people in their party
27          Console.WriteLine("The total in your party is: " + total);
28
```

<table>
<tr><th>Question number</th><th>Answer</th><th>Additional guidance</th><th>Mark</th></tr>
<tr><td>2(b)</td><td>For up to two tests

Award 1 mark for an appropriate validation test and 1 mark for an item of test data that would fail the given test.

<table>
<tr><th>Validation test</th><th>Invalid data</th></tr>
<tr><td>Is length = 5?</td><td>ABC-123</td></tr>
<tr><td>Is hyphen in the middle?</td><td>AB1-2</td></tr>
<tr><td>Are characters 1 and 2 letters?</td><td>12-34</td></tr>
<tr><td>Are characters 4 and 5 digits?</td><td>AB-MP</td></tr>
</table>
</td><td></td><td>(4)</td></tr>
</table>

<table>
<tr><th>Question number</th><th>Answer</th><th>Additional guidance</th><th>Mark</th></tr>
<tr><td>2(c)</td><td>
Award 1 mark for each logical test.
<table>
<tr><th>Condition</th><th>Output message</th><th>Logical test</th></tr>
<tr><td>Attendance is at least 1500</td><td>Sufficient profit made this week</td><td rowspan="2">(attendance >= 1500) or (income >= 45000)</td></tr>
<tr><td>Income is at least 45000</td><td>Sufficient profit made this week</td></tr>
<tr><td>Attendance is at least 750; income is at least 22500</td><td>Income in line with attendance this week</td><td>(attendance >= 750) and (income >= 22500)</td></tr>
<tr><td>Attendance is fewer than 500</td><td>Attendance is very low this week
Contact fan club</td><td>(attendance < 500)</td></tr>
<tr><td>All other inputs</td><td>Possible accounting error</td><td>The print statement needs to be in the ‘else’ block</td></tr>
</table>
Python

```
if (attendance >= 1500) or (income >= 45000):
    print ("Sufficient profit made this week")
elif (attendance >= 750) and (income >= 22500):
    print ("income in line with attendance this week")
elif (attendance < 500):
    print ("Attendance is very low this week.  Contact the fan club.")
else:
    print ("Possible accounting error.")
```

Java

```
if ((attendance >= 1500) || (income >= 45000))
    System.out.println("Sufficient profit made this week");
else if ((attendance >= 750) && (income >= 22500))
    System.out.println("income in line with attendance this week");
else if (attendance < 500)
    System.out.println("Attendance is very low this week.  Contact the fan club.");
else
    System.out.println("Possible accounting error.");
```

</td><td>Candidates are required to open the file Q02c in the code editor. Amended code should be saved as Q02cFINISHED.

Do not penalise candidates who attempt more than the stated requirements.</td><td>(4)</td></tr>
</table>

	C#
	`if ((attendance >= 1500) || (income >= 45000))`
	`  Console.WriteLine("Sufficient profit made this week");`
	`else if ((attendance >= 750) && (income >= 22500))`
	`  Console.WriteLine("income in line with attendance this week");`
	`else if (attendance < 500)`
	`  Console.WriteLine("Attendance is very low this week.  Contact the fan club.`
	`else`
	`  Console.WriteLine("Possible accounting error.");`		

Question number	Answer	Additional guidance	Mark
3(a)	Any **one** from: • a subprogram can be written once (1) and called many times (1) • a subprogram can be debugged once (1) and called many times (1) • subprograms can be collected into libraries (1), which can be used by other programs (1).		**(2)**

Question number	Answer	Additional guidance	Mark
3(b)(i)	**Python** time.sleep/print **Java** Thread.sleep/System.out.println **C#** System.Threading.Thread.Sleep/Console.WriteLine	Candidates are required to open the file Q03b in the code editor. Accept clear reference to sleep and print built-in subprograms.	**(1)**

Question number	Answer	Additional guidance	Mark
3(b)(ii)	toCelsius/toFahrenheit/waitTenSeconds/waitTime	Candidates are required to open the file Q03b in the code editor. These are the same across all languages. Accept clear reference to subprogram name.	(1)

Question number	Answer	Additional guidance	Mark
3(b)(iii)	**Python** inTemp/inSeconds **Java and C#** inTemp/inMilliseconds	Candidates are required to open the file Q03b in the code editor. Accept clear reference to parameter name.	(1)

Question number	Answer	Additional guidance	Mark
3(b)(iv)	waitTenSeconds	Candidates are required to open the file Q03b in the code editor. This is the same across all languages. Accept clear reference to subprogram name.	(1)

Question number	Answer	Additional guidance	Mark
3(b)(v)	**Python** inTemp/celsius/fahrenheit/inSeconds **Java and C#** inTemp/celsius/fahrenheit/inMilliseconds	Candidates are required to open the file Q03b in the code editor. Accept clear reference to variable name.	**(1)**

Question number	Answer	Additional guidance	Mark
3(b)(vi)	theDate	Candidates are required to open the file Q03b in the code editor. This is the same across all languages. Accept clear reference to variable name.	**(1)**

Question number	Answer	Additional guidance	Mark
3(b)(vii)	The subprogram (being called on this line) is missing a return statement.	Candidates are required to open the file Q03b in the code editor.	**(1)**

<table>
<tr><th>Question number</th><th>Answer</th><th>Do not accept</th><th>Additional guidance</th><th>Mark</th></tr>
<tr><td>3(c)</td><td>Award one mark for each of:

• opening ‘Cities.txt’ for reading only (1)
• opening/creating ‘Numbered.txt’ for writing only (1)
• using for-each/while not EOF to read each line into variable ‘theLine’ (1)
• incrementing the count of lines read (1)
• constructing the output string/coercion as required (1)
• writing new lines to the output file (1)
• closing at least one of the text files (1).</td><td>Data structures such as an array or a list.</td><td rowspan="2">Candidates are required to open the file Q03c in the code editor. Amended code should be saved as Q03cFINISHED.</td><td rowspan="2">(7)</td></tr>
<tr><td colspan="2">Python


```
# Open "Cities.txt" as input
theFile = open ("Cities.txt", "r")

# Open "Numbered.txt" as output
outFile = open ("Numbered.txt", "w")

# Use a for/each loop to read each line of
# the input file into a variable named 'theLine'
for theLine in theFile:

    # Increment the line count
    count = count + 1

    # Add the line number to the front of the line followed by a space
    theLine = str(count) + " " + theLine

    # print the line to the display
    print (theLine)

    # Write the new line to the output file
    outFile.writelines (theLine)

# Close the input file
theFile.close()

# Close the output file
outFile.close()
```

</td></tr>
</table>

Java

```
20          // Open "Cities.txt" as input
21          theFile = new Scanner(new BufferedReader(new FileReader("cities.txt")));
22          // Open "Numbered.txt" as output
23          outFile = new PrintWriter("Numbered.txt", "UTF-8");
24          // Use loop to read each line of
25          // the input file into a variable named 'theLine'
26          while (theFile.hasNextLine())
27          {
28              theLine = theFile.nextLine();
29              // Increment the line count variable
30              count++;
31              // Add the line number to the front of the line
32              // followed by a space
33              theLine = (Integer.toString(count) + " " + theLine);
34              // print the line to the display
35              System.out.println(theLine);
36              // Write the new line to the output file
37              outFile.println(theLine);
38          }
39
40          // Close the input file
41          theFile.close();
42          // Close the output file
43          outFile.close();
```

C#

```
13          // open Cities.txt as input
14          System.IO.StreamReader fileReader = new System.IO.StreamReader("Cities.txt");
15          // open Numbered.txt as output
16          System.IO.StreamWriter fileWriter = new System.IO.StreamWriter("Numbered.txt");
17          // use loop to read each line of the input file into a variable named theLine
18          while (fileReader.Peek() >= 0)
19          {
20              theLine = fileReader.ReadLine();
21              // increment the line count variable
22              count = count + 1;
23              // add the line number to the front of the line followed by a space
24              theLine = Convert.ToString(count) + " " + theLine;
25              // print the line to the display
26              Console.WriteLine(theLine);
27              // write new line to the output file
28              fileWriter.WriteLine(theLine);
29          }
30          // close the input file
31          fileReader.Close();
32          // close the output file
33          fileWriter.Close();
34          Console.ReadKey();
```

Question number	Answer	Additional guidance	Mark
4(a)	Any **one** from: • a step-by-step procedure (which if followed precisely with a given input produces a predictable output) • a list of instructions followed in sequence (to solve a problem) • a process or set of rules to be followed (to achieve a predictable result).		**(1)**

Question number	Answer	Additional guidance	Mark
4(b)(i)	7 | 26 | 21 | 28 | 18 | 16 | 9 | 34		**(1)**

Question number	Answer	Additional guidance	Mark
4(b)(ii)	7		**(1)**

Question number	Answer	Additional guidance	Mark
4(b)(iii)	6		**(1)**

Question number	Answer	Additional guidance	Mark
4(c)(i)	Any **one** from: • requires many passes to complete the sort • requires many comparisons/every number is compared every single pass.		**(1)**

Question number	Answer	Additional guidance	Mark
4(c)(ii)	Top/highest/right-most/last		**(1)**

<table>
<tr><th>Question number</th><th>Answer</th><th>Additional guidance</th><th>Mark</th></tr>
<tr><td>5(a)</td><td>Award 1 mark for each correct pass of the loop.

<table>
<tr><th>target</th><th>rs</th><th>rm</th><th>r</th><th></th></tr>
<tr><td>4</td><td>0</td><td>0</td><td>1</td><td>(1)</td></tr>
<tr><td>4</td><td>1</td><td>0</td><td>1</td><td></td></tr>
<tr><td>4</td><td>1</td><td>1</td><td>1</td><td></td></tr>
<tr><td>4</td><td>1</td><td>1</td><td>2</td><td>(1)</td></tr>
<tr><td>4</td><td>4</td><td>1</td><td>2</td><td></td></tr>
<tr><td>4</td><td>4</td><td>2</td><td>2</td><td></td></tr>
<tr><td>4</td><td>4</td><td>2</td><td>3</td><td>(1)</td></tr>
<tr><td>4</td><td>9</td><td>2</td><td>3</td><td></td></tr>
<tr><td>4</td><td>9</td><td>3</td><td>3</td><td></td></tr>
<tr><td>4</td><td>9</td><td>3</td><td>4</td><td>(1)</td></tr>
<tr><td>4</td><td>16</td><td>3</td><td>4</td><td></td></tr>
<tr><td>4</td><td>16</td><td>0</td><td>4</td><td></td></tr>
<tr><td>4</td><td>16</td><td>0</td><td>5</td><td>(1)</td></tr>
</table></td><td>Candidates are required to open the file Q05a in the code editor.

Penalise each mathematical error once and then follow through.</td><td>(5)</td></tr>
</table>

Question number	Answer	Additional guidance	Mark
5(b)	• Accept user input of total spend (1) • Coercion of input (to a numerical data type) (1) • Correct logic for totalSpend greater than 300 leads to printing correct output message (1) • Correct logic for totalSpend greater than 0 leads to printing correct output message (1) • Correct logic for all other input leads to printing correct output message. (1)	Candidates are required to open the file Q05b in the code editor. Amended code should be saved as Q05bFINISHED.	**(5)**

	Python
	
	`# Write your code below this line`
	`totalSpend = int (input ("What is your total spend?"))`
	`if (totalSpend > 300):`
	`    print ("Discount is 10%")`
	`elif (totalSpend > 0):`
	`    print ("No discount")`
	`else:`
	`    print ("Invalid input")`		

Java

```
1  package q05b;
2  import java.util.Scanner;
3  // Write your code here
4  public class Q05b
5  {
6      public static void main(String[] args)
7      {
8          Scanner input = new Scanner(System.in);
9          int totalSpend = 0;
10
11         System.out.println ("What is your total spend?");
12         totalSpend = input.nextInt();
13
14         if (totalSpend > 300)
15             System.out.println ("Discount is 10%");
16         else if (totalSpend > 0)
17             System.out.println ("No discount");
18         else
19             System.out.println ("Invalid input");
20     }
21 }
```

C#

```
using System;
// Write your code here

namespace Q05b_cs
{
    class Program
    {
        static void Main(string[] args)
        {
            int totalSpend = 0;

            Console.WriteLine("What is your total spend?");
            totalSpend = Convert.ToInt32(Console.ReadLine());

            if (totalSpend > 300)
                Console.WriteLine("Discount is 10%");
            else if (totalSpend > 0)
                Console.WriteLine("No discount");
            else
                Console.WriteLine("Invalid input");
        }
    }
}
```

Question number	Answer	Additional guidance	Mark
6	Award **one** mark for each of the following points up to a maximum of **11** marks: • Initialise 'make artist label' loop (1) • Initialise 'find youngest person' loop to cycle through all artists given (1) • Identify / extract initials of artist (1) • Combine/concatenate initials with year of birth to create label (1) • Add the label to the **theLabels** data structure (1) • Display the label for the artist (1) • Initialise the year of birth variable (for finding youngest artist e.g. maxDate) **or** initialise oldest person variable (to hold oldest person e.g. maxPerson) (1) • Check date of birth with year of birth variable (1) • Assign younger year to maxDate if necessary (1) • Assign older year to maxPerson if necessary (1) • Display the name and year of birth of artist identified (1)		**(11)**

Award up to a maximum of **nine** marks using the levels-based mark scheme below.				
Band 0	**Band 1 (1-3 marks)**	**Band 2 (4-6 marks)**	**Band 3 (7-9 marks)**	**Mark**
No rewardable content	Little attempt to decompose the problem into component parts	Some attempt to decompose the problem into component parts	The problem has been decomposed into component parts	
	Some parts of the logic are clear and appropriate to the problem	Most parts of the logic are clear and mostly appropriate to the problem	The logic is clear and appropriate to the problem	
	Some appropriate use and manipulation of data types, variables, data structures and program constructs	The use and manipulation of data types, variables, data structures and program constructs is mostly appropriate	The use and manipulation of data types, variables and data structures and program constructs is appropriate	
	Parts of the code are clear and readable	Code is mostly clear and readable	Code is clear and readable	
	Finished program will not be flexible enough with other data sets or input	Finished program will function with some but not all other data sets or input	Finished program could be used with other data sets or input	
	The program meets some of the given requirements	The program meets most of the given requirements	The program fully meets the given requirements	**(9)**

Example solutions

Python

```
# Make the artist labels
for person in theArtists:
    newRecord = person[1][0] + person[0][0] + str(person[2])
    theLabels.append (newRecord)
print ("The new userIDs are: ", theLabels)

# Find and print the youngest person and their birthdate
maxDate = 0
for person in theArtists:
    if person[2] > maxDate:
        maxDate = person[2]
        maxPerson = person
print (maxPerson[0], maxPerson[1], "is youngest", str(maxPerson[2]))
```

Java

```
        // Make the artist labels
        for (int i = 0; i < theArtists.length; i++) {
            String newRecord = String.valueOf(theArtists[i][1].charAt(0))
                    + String.valueOf(theArtists[i][0].charAt(0))
                    + theArtists[i][2];
            theLabels.add (newRecord);
            System.out.println(newRecord);
            System.out.println("The new userIDs are: " + theLabels.get(theLabels.size() - 1));
        }
        // Find and print the youngest person and their birthdate
        int maxDate = 0;
        String maxPerson = "";
        for (String[] person : theArtists) {
            if (Integer.parseInt(person[2]) > maxDate) {
                maxDate = Integer.parseInt(person[2]);
                maxPerson = person[0] + " " + person[1];
            }
        }
        System.out.println(maxPerson + " is youngest " + maxDate);
```

C#

```
for (int i = 0; i < theLabels.Length; i++)
{
    String newRecord = Convert.ToString(theArtists[i,1][0]) +
            Convert.ToString(theArtists[i,0][0]) +
            theArtists[i,2];
    theLabels[i] = newRecord;
    Console.WriteLine(newRecord);
    Console.WriteLine("The new userIDs are: " + theLabels[i]);
}
// Find and print the youngest person and their birthdate
int maxDate = 0;
String maxPerson = "";
for (int i =0; i< theLabels.Length;i++)
{
    if (Convert.ToInt32(theArtists[i,2]) > maxDate)
    {
        maxDate = Convert.ToInt32(theArtists[i,2]);
        maxPerson = theArtists[i,0] + " " + theArtists[i,1];
    }
}
Console.WriteLine(maxPerson + " is youngest " + maxDate);
Console.ReadKey();
```

Pseudocode reference

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

Subprograms		
Syntax	**Explanation of syntax**	**Example**
<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.

For information about Edexcel, BTEC or LCCI qualifications
visit qualifications.pearson.com

Edexcel is a registered trademark of Pearson Education Limited

Pearson Education Limited. Registered in England and Wales No. 872828
Registered Office: 80 Strand, London WC2R 0RL
VAT Reg No GB 278 537121

Getty Images: Alex Belmonlinsky

ISBN 978-1-4469-4932-0