Pearson Edexcel

Mark Scheme (Pre-Standardisation)

Summer 2018

GCSE Computer Science (1CP1/02)
Paper 2: Application of Computational Thinking

Question Number	Answer	Additional Guidance	Mark
1(a)	Any **two** from • Date of birth (1) • Employee number (1) • Permitted areas (1) • Department (1) • Salary (1) • Address (1) • Job role (1) • Badge number (1) • National Insurance number (1) • Location/depot (1) • Uniform information (1)	• Accept meaningful variable names tied to the scenario, e.g. EmployeeID, DOB, etc. • Do not accept age • Start date is in the question • Accept other examples appropriate to the scenario	**2**

Question Number	Answer	Additional Guidance	Mark
1(b)	• Calculation showing length of employment in years (1) • Subtracting 2 for the first two years (1) • Multiplying by 0.5 days per year (1) • Adding in the 10 days basic holiday entitlement (1) Example: • 10 + (((currentYear – startYear) - 2) x 0.5)	• Units not required • Brackets not required if order of precedence rules are followed (BIDMAS)	**4**

<table>
<tr><th>Question Number</th><th>Answer</th><th>Additional Guidance</th><th>Mark</th></tr>
<tr><td>1(c)</td><td>One mark for each cell.

<table>
<tr><th>Input(s)</th><th>Process</th><th>Output(s)</th></tr>
<tr><td>Location of card reader

Numeric code from magnetic strip</td><td>Find out if this employee can go through this door</td><td>True, if entry permitted

False, if entry not permitted</td></tr>
<tr><td>Cost of canteen meal

Numeric code from magnetic strip</td><td>Pay for canteen meal / subtract price of meal from balance / check if enough money in account</td><td>New balance, if paid

Error, if not enough in account</td></tr>
<tr><td>Top-up amount

Numeric code from magnetic strip</td><td>Add amount of top-up to make new balance</td><td>New balance</td></tr>
<tr><td>Numeric code from magnetic strip / account number</td><td>Check account balance</td><td>New balance</td></tr>
</table></td><td>• Award as long as meaning is discernible</td><td>3</td></tr>
</table>

Question Number	Answer	Additional Guidance	Mark
2 (a)	In this exact order: • South depot / South (1) • North depot / North (1) • Unknown code / Unknown (1)	Ignore spelling Ignore extraneous text after a correct response	**3**

Question Number	Answer	Additional Guidance	Mark
2 (b)	One mark for each check and one for a linked example of test data that fails **the** test Type check (1) Data (1) Length check (1) Data (1) Presence check (1) Code is blank (1) Lookup check (1) Explanation of data not in pre-existing lists (1)	• Data must follow test • Quotes not required • Data must fail the indicated test and not several tests • Do not interpret blank cell as an answer to presence check	**4**

<table>
<tr><th>Question Number</th><th>Answer</th><th>Additional Guidance</th><th>Mark</th></tr>
<tr><td>3 (a)(i)</td><td>
<table>
<tr><th>Programming construct</th><th>Line number(s)</th></tr>
<tr><td>A condition controlled loop</td><td>One from:
• 8 (1)
• 8-18 (1)</td></tr>
<tr><td>A comment</td><td>One from:
• 2 (1)
• 7 (1)
• 20 (1)</td></tr>
<tr><td>A selection construct</td><td>One from:
• 12-17 (1)
• 12,15,17 (1)</td></tr>
<tr><td>A subprogram call</td><td>One from:
• 9 (1)
• 10 (1)
• 14 (1)
• 16 (1)
• 21 (1)
• 24 (1)</td></tr>
</table>
</td><td>• Accept words instead of numbers</td><td>4</td></tr>
</table>

Question Number	Answer	Additional Guidance	Mark
3 (a)(ii)	Boolean (1)	Ignore spelling and extraneous text after a correct response	**1**

Question Number	Answer	Additional Guidance	Mark
3 (a)(iii)	One from: • Real (1) • Float (1) • Double (1)	Ignore spelling and extraneous text after a correct response	**1**

Question Number	Answer	Additional Guidance	Mark
3 (b)	One mark for each correct cell. **currentCount** | **i** | **LENGTH(weekNumber)** | **percentage** | **staffRates[i]** | **neededStaff** 300 | 2 | 6 | 1.15 | 115 | 345	• If the calculation is wrong on line 22 (percentage), award a follow through to line 23 (neededStaff = 300 * percentage) • Award equivalent expressions	**6**

Question Number	Answer	Additional Guidance	Mark
3 (c)	199 (1)		**1**

Question Number	Answer	Additional Guidance	Mark
4(a)	Logic / logical (1) Example: Logical / arithmetic (1) as logical is first Arithmetic / logical (0).	Ignore spelling and extraneous text following a correct response	**1**

Question Number	Answer	Additional Guidance	Mark
4(b)	One mark for each cell. Marks can be awarded independently. **Error** / **Correction** Line 11: The loop never executes (1) because found is initialised to False / because of found = True — WHILE (found = False) AND … / Accept 'found = False' Line 18: It prints the wrong item type (1) / It prints the item type one place to the right (1) of the correct one / [i + 1] — SEND typeItem[i]) TO DISPLAY	• Accept prose description of error for Error column • Ignore any copying errors • Accept any discernible notation for correction	**4**

Question Number	Answer	Additional Guidance	Mark
5(a)	Line 2: inWidth, inHeight, inLength (1) – 3 meaningful variable names, order must match line 19 Line 6: inWidth * inHeight * inLength (1) – must match names from line 2, but order of calculation doesn't matter Line 8: volume (1) – only possible response Line 19: itemWidth, itemHeight, itemLength (1) – only possible names because they are used in the main program, any order See next page for an example.	• Ignore spelling mistakes and syntax as long as logic is discernible	**4**

```
FUNCTION calcVolume (inWidth, inHeight, inLength)

BEGIN FUNCTION

    volume = inWidth * inHeight * inLength

    RETURN (volume)

    END FUNCTION

SEND "Enter width" TO DISPLAY
RECEIVE itemWidth FROM (INTEGER) KEYBOARD
SEND "Enter height" TO DISPLAY
RECEIVE itemHeight FROM (INTEGER) KEYBOARD
SEND "Enter length" TO DISPLAY
RECEIVE itemLength FROM (INTEGER) KEYBOARD

SET itemVolume TO calcVolume (itemWidth, itemHeight, itemLength)

SEND "Volume is " & itemVolume TO DISPLAY
```

Question Number	Answer	Additional Guidance	Mark
5(b)	Any one of: • A function returns a value (1) • Without returning a value, it is not a function (1) • A procedure does not return a value (1)		**1**

Question Number	Answer	Additional Guidance	Mark
5(c)	volume	Ignore spelling and extraneous text after a correct response	**1**

Question Number	Answer	Additional Guidance	Mark
5(d)	Any two from: • Reduced coding time / code is quicker to write / code can be reused (1) • Reduced debugging time / library code is bug-free / reduced testing time (1) • Library may have more complex functionality (1) than the programmer has understanding or capability		**2**

Question Number	Answer	Additional Guidance	Mark
6(a)(i)	Random order (1) means that all items will have to be looked (1) at to find all occurrences Because the data is not sorted (1), the algorithm has to look at every item (1) in the list to find all occurrences.		**2**

Question Number	Answer	Additional Guidance	Mark
6(a)(ii)	The loop (1) means that all records will have to be looked at even if there are no numbers > 300 in the list (1)		**2**

Question Number	Answer	Additional Guidance	Mark

6(a)(iii)	Any one from:
	
	• Sort the list (1)
	• Direction of search (ascending, descending [implied or explicit]) matches order of sort (1) E.g.
	  ○ Ascending sort order requires the search to start from the end of the set
	  ○ Descending sort order requires the search to start from the beginning of the set.
	• Test for stopping for found item <300 (1)
			3

<table>
<tr><th>Question Number</th><th>Answer</th><th>Additional Guidance</th><th>Mark</th></tr>
<tr><td>6(b)</td><td>

Appropriate and meaningful column headings (1)
Two rows of data whose type and values are sensible for the column headings (1)

<table>
<tr><td>Van ID</td><td>Monday</td><td>Tuesday</td></tr>
<tr><td>8765</td><td>310</td><td>379</td></tr>
<tr><td>212</td><td>187</td><td>231</td></tr>
</table>
</td><td>

Question requires a 2D array, so data types for lower 2 rows have to be homogeneous

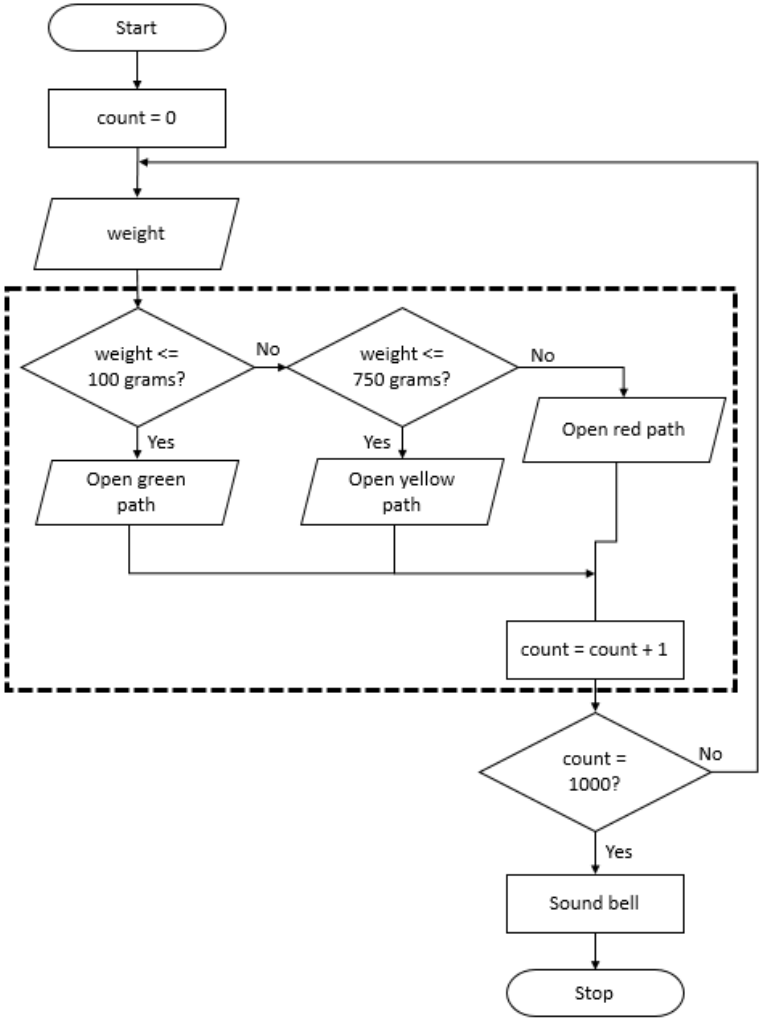
</td><td>2</td></tr>
</table>

<table>
<tr><th>Question Number</th><th>Answer</th><th>Additional Guidance</th><th>Mark</th></tr>
<tr><td>6(c)</td><td>
Any two from:

1. Drivers may feel it is an invasion of privacy (1) because they could be tracked during work hours when they are not delivering items (1).
2. Drivers may not feel it is acceptable to be ranked/compensated based on tracking data (1) because unknown events (road works, etc.) will affect the tracking data (1).
3. Drivers may be tracked by customers (1) who may be annoyed by perceived delays (1).
4. Drivers may worry that their personal data (home address) may not be secure (1) and susceptible to unauthorised access (1).
5. Drivers may want to use the vans for their own purposes (1) outside of work hours and may not want this activity tracked (1).
</td><td></td><td>4</td></tr>
</table>

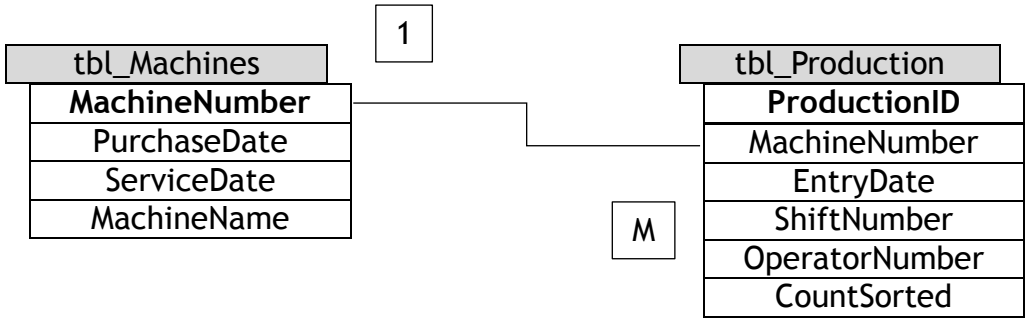
Question Number	Answer	Additional Guidance	Mark

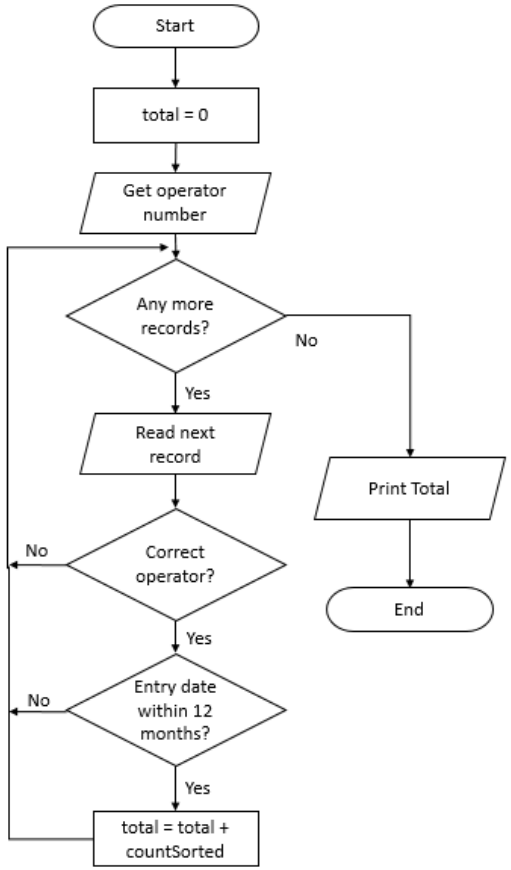
7 (a)	A completed flow chart that includes:
	• Decision diamonds (1)
	  ○ Directly follow one after the other from lowest to highest
	  ○ Label for flow between them must be 'No'
	  ○ Flow is unbroken by other instructions as this changes from if/elseif/else to sequence
	• Coloured paths
	  ○ Green path directly follows <=100 decision with label 'Yes' (1)
	  ○ Yellow path directly follows <=750 decision with label 'Yes' (1)
	  ○ Red path directly follows <=750 decision with label 'No' (1)
	• Flow
	  ○ Three distinct branches which join back into the flow of the loop construct (i.e. do not join back to each other, decision symbols, end, or additional symbols which have not been provided) (1)
	• Increment
	  ○ Single box positioned before any branches / positioned as the last process after all branches (1)
		• Different ordering and layout may be used
		• Each symbol is to be used only once (in question) and no other symbols should be used
		• Do not penalise lack of arrows on lines
		• Do not penalise inaccurate drawing of symbols
		• There should be no looping or flow which degrades what is an if/elseif/else construct to multiple if constructs
		• A process box can have more than one input (e.g. count=count+1), but not a decision box
			6


Start
count = 0
weight
weight <= 100 grams?
No
weight <= 750 grams?
No
Open red path
Yes
Open green path
Yes
Open yellow path
count = count + 1
count = 1000?
No
Yes
Sound bell
Stop

Question Number	Answer	Additional Guidance	Mark
7(b)(i)	• To **uniquely** identify an entity / a record in a table (1)		**1**

Question Number	Answer	Additional Guidance	Mark
7(b)(ii)	• MachineNumber (1)	Ignore spelling and extraneous text after a correct response	**1**

Question Number	Answer	Additional Guidance	Mark
7(b)(iii)	Line to connect the correct fields. (1) Correct labelling of the degree of the relationship. (1) 	• Other notation for relationships acceptable 1:Many 1:∞ Crows feet:  • Keys must be clearly connected, not just the tables	**2**

Question Number	Answer	Additional Guidance	Mark
7(b)(iv)		There are a maximum of 3 marks for functionality. There are a maximum of 3 marks for accuracy of notation. The input/output symbol can be replaced by the process symbol. There are many different solutions, but the flowchart must attempt to solve the problem Indicative content Initialisation of running total; Incrementing running total; Checking if this record is of interest; Handling end of data structure; Printing total as output.	**6**

Aspect of Solution	**Marks**			
	0	**1**	**2**	**3**
Functionality	No rewardable content	There are significant errors in logic, leading to an overall solution that is non-functional.	There are minor errors in logic, leading to an overall solution that is not completely functional.	There are no errors in logic, leading to an overall solution that is fully functional.
Accuracy of notation	No rewardable content	Notation follows a broadly unrecognisable convention that is applied inconsistently, although aspects of it are discernible.	Notation follows a recognisable convention, which is broadly discernible but is applied inconsistently.	Notation follows a recognisable convention and is applied consistently throughout.

<table>
<tr><th>Question Number</th><th>Answer</th><th>Additional Guidance</th><th>Mark</th></tr>
<tr><td>8</td><td>
<table>
<tr><th rowspan="2">Aspect of Solution</th><th></th><th>Marks</th><th></th><th></th></tr>
<tr><th>0</th><th>1</th><th>2</th><th>3</th></tr>
<tr><td>Functionality</td><td>No awardable content</td><td>There are significant errors in logic, leading to an overall solution that is non-functional.</td><td>There are minor errors in logic, leading to an overall solution that is not completely functional.</td><td>There are no errors in logic, leading to an overall solution that is fully functional.</td></tr>
<tr><td>Accuracy of Notation</td><td>No awardable content</td><td>Notation follows a broadly unrecognisable convention that is applied inconsistently, although aspects of it are discernible.</td><td>Notation follows a recognisable convention, which is broadly discernible but is applied inconsistently.</td><td>Notation follows a recognisable convention and is applied consistently throughout.</td></tr>
<tr><td>Efficiency, Appropriateness, and Accuracy of Solution</td><td>No awardable content</td><td>There are significant errors in the selection and accurate use of appropriate techniques.</td><td>Techniques have been selected and used with some accuracy, although the techniques may not be the most appropriate.</td><td>Techniques have been selected and used accurately and appropriately throughout to demonstrate an efficient solution.</td></tr>
</table>
There are a maximum of 3 marks for functionality.
There are a maximum of 3 marks for accuracy of notation.
There are a maximum of 3 marks for efficiency, appropriateness, and accuracy of solution.
Each row is awarded independently.
</td><td></td><td>9</td></tr>
</table>

Question 8 - Pseudocode Example:

```
2  SET count TO 0
3  SET maxWeight TO 0
4  SET minWeight = 9999
5
6  SEND "Enter the weight for the item, 0 to quit: " TO DISPLAY
7  RECEIVE weight (STRING) FROM KEYBOARD
8  WHILE (NOT (weight = 0)):
9      IF (weight < 0) THEN
10         SEND "Error.  No negatives allowed" TO DISPLAY
11     ELSE
12         SET count TO count + 1
13         IF (weight > maxWeight) THEN
14             SET maxWeight TO weight
15         IF (weight < minWeight) THEN
16             SET minWeight TO weight
17         ENDIF
18     ENDIF
19     SEND "Enter the weight for the item, 0 to quit: " TO DISPLAY
20     RECEIVE weight (STRING) FROM KEYBOARD
21 END WHILE
22 SEND "Count = " & count TO DISPLAY
23 SEND "Heaviest = " & maxLetterWeight TO DISPLAY
24 SEND "Lightest = " & minWeight TO DISPLAY
```

Initialisation of variables
minWeight should be high so that it will be set with the first value entered.

Condition controlled loop required

Validation for negative numbers

First pass through the loop means that minWeight and maxWeight will be the same number. If more than one number is entered, then an if/else would also work, but not in all cases.