Pearson
Edexcel

Examiners' Report
Principal Examiner Feedback

Summer 2019

Pearson Edexcel International GCSE in
Computer Science (4CP0)
Paper 02: Application of Computational Thinking

Grade Boundaries

Grade boundaries for all papers can be found on the website at: https://qualifications.pearson.com/en/support/support-topics/results-certification/grade-boundaries.html

Summer 2019
Publications Code 4CP0_02_1906_ER

This report is split into two sections: General Comments and Specific Comments. In the Specific Comments, there will be comments about the candidates' responses to the written and coding questions.

GENERAL COMMENTS

This was the first series of the Specification of Pearson's International GCSE Computer Science.

There were approximately 450 candidates for the specification in this series. The large majority of candidates attempted all questions and the three hours allowed for the examination did not seem to be an issue for most candidates.

The format of the question paper is a combination of written theoretical questions about computational thinking and practical coding tasks. It is intended that the structure of the paper is such that demand increases through each question and through the paper as a whole. The approximate split, in terms of marks, is approximately 40% written responses and 60% coding responses. There will normally be 5, 6 or 7 questions, with the last question in each series an extended coding exercise intended to allow candidates to demonstrate their knowledge, skills and understanding of computational thinking applied to a problem. The questions requiring written responses and coding responses were interspersed to allow candidates time to be looking away from computer screens.

Candidates are required to complete the coding exercises using one of three programming languages: C sharp (C#), Java or Python. For this series, the majority of candidates submitted work using Python as their programming language. A small number of centres had different candidates using one of two programming languages.

It is expected that candidates will be familiar with how to enter code that is exemplified in the pseudocode resource that forms Appendix 5 of the specification. A copy of this pseudocode resource was (and will be) included in the question paper.

Most centres submitted the candidates' work in the appropriate manner with the scripts and the coding responses in the same envelope. For the most part, candidates' work was correctly identified. However, some centres identified the candidates' work by the name of the candidate and not according to the instructions in the ICE document (**I**nformation for the **C**onduct of **E**xaminations). This document (updated each year) is usually available on the Pearson website early in the year of the examination. A few centres had to be contacted because the script envelope did not contain one of either the written responses or electronic coding files (on a CD/DVD or a USB drive).

Due to the format of the question paper, the mark scheme is arranged so that the questions with written responses are grouped at the start of the scheme, followed by the questions with coding responses. Examples of coding that meet the requirements of the question paper were grouped at the end (in programming language order).

Later in the year, some further material will be available to provide exemplar materials with commentaries. This will be available on the Pearson website on the pages for International GCSE Computer Science.

SPECIFIC COMMENTS

Written response questions

- The multiple-choice questions included in the question paper were generally well answered with the majority of candidates scoring well.
    - In question 1(a), many candidates who answered incorrectly thought that 'string' was more appropriate than 'real' for the value 4.5.
    - In 4(a), the most common incorrect answers were 'shift' and 'plaintext'.

- Single mark questions were also generally well answered.
    - In 1(b), the most frequent reasons for loss of marks was for vagueness in the response or repeating the stem of the question (e.g. a logic error is an error in the logic.).
    - Very few candidates failed to score the mark for 1(d).
    - In 2(b), it would seem that some candidates did not use the code given to answer parts of the question. These often gave incorrect responses to parts (i) and (iv) as they tried to calculate the answers themselves.
    - Most candidates were aware of BODMAS or BIDMAS for part 2(b)(ii)
    - In 2(c), line numbers were correctly identified by a number of candidates in parts (i) and (ii). Some added the actual code to demonstrate their understanding of the code.
    - 2(c) parts (iii) and (iv) were less well done with candidates often including the incorrect responses such as 'check' and 'outMessage'

- Multiple mark questions were generally less well answered with answers often not gaining full marks due to a lack of expansion of the response.
    - 1(e) was generally well answered with most candidates being able to identify the difference between local and global variables.
    - 1(g) was probably the least well answered written question on the paper. Whilst candidates showed some understanding of the need for validation tests, responses were often too vague to gain credit. What was expected is that candidates would apply computational thinking to the question and identify tests that could be coded. The examples of erroneous data often failed more than one criterion given in the question.
    - 2(b)(iii) was usually well answered. Whilst the correct response for the first part was 'Real', 'float' and 'double' were also accepted (though these are not mentioned in the specification as data types).
    - 2(c)(v): most candidates were able to use the code to obtain the correct response. However, others did not realise that 'Thank you' was not the text returned by the subprogram.
    - 2(c)(vi): most candidates were able to respond correctly in identifying the purpose of the subprogram in the code. Some responded with a generic answer about the purpose of subprograms. These responses were given

some credit but not full credit as they had not answered the question given.

- 3(b) – many responses identified that a merge sort required lots of comparisons and iterations, but few candidates gained credit for mentioning the memory/storage requirements of the algorithm
- 4(b) was generally answered well, with very few candidates failing to score all 3 marks.
- 4(c) was also generally answered well, though some were, perhaps, confused by the inclusion of a space in the character set.

Coding response questions

Generally, examiners found that candidates responded very well to the coding challenges presented in the question paper. There were many candidates who scored close to full marks on these tasks.

1(c) Most candidates scored all three marks for the correction of errors in the short section of code presented. Where candidates did lose marks, it was often in not initialising the constantValue variable with the value of 7

1(f) Again, generally well done by most candidates. Marks lost were due to incorrect relational operators (e.g. > rather than >=) and incorrect logic operator especially in the second statement (using the coding for 'OR' rather than 'AND')

2(a) Candidates were required to 'translate' the pseudocode into the programming language. Most candidates scored well. Errors were often caused by incorrect application of the random generator. Other errors were introduced in the WHILE statement where an incorrect comparison was coded. As mentioned earlier, candidates need to be able to translate into the programming code of their choice the structures presented within the pseudocode. Several incorrect responses only copied the pseudocode into the programming environment including the capitalisation given in the question paper (e.g. SET counter TO 1).

3(a) Many candidates found this task particularly difficult with often a lack of ability in opening and closing text files correctly and so they were then unable to check the contents of the addresses for the '@' symbol. Many incorrect responses also checked whether the complete address was equal to the '@' symbol rather than whether the address contained the symbol. Some others checked each character and wrote the address to the file each time the character was not the '@' symbol.

3(c) This was designed as a simple task to allow candidates to demonstrate their knowledge and skill in writing a program without any of the structure provided in earlier questions. There were some very good responses to this task with some going beyond a single iteration by introducing a loop to continue the program until an invalid number was entered.

5 Examiners reported that there were some excellent responses to this extended question. Candidates were given a data set of some library members and the number of books they had read. They were asked to create code to analyse the data by totalling the number of books read and calculate the average number. They were also asked to find the identification code of readers of less than ten books and the details of the top three readers. Examiners reported that there were many candidates who produced very elegant and efficient programs to carry out the task.

Where some candidates did lose marks was in not using coding to do the calculations and identification of individual readers. Instead they had carried out a manual search and then printed out the results of the search.

Others had not allowed for alternative data sets that might contain a different number of library records.

Various alternative approaches to identifying the relevant information were seen. Some used sorting algorithms on the original data set, others did a compare and replace / append relevant values.

Some candidates included the possibility that the data set might contain multiple pupils winning gold, silver or bronze medals. Those candidates who did not do this did not lose marks.

Pearson Education Limited. Registered company number 872828
with its registered office at 80 Strand, London, WC2R 0RL, United Kingdom