

General:

The question paper was, generally speaking, well answered. It contained both easy and more difficult questions. A recurring problem was that candidates did not always pay attention to the small details in questions and thus lost marks.

Section A: Database Application

Question 1:

1.1 -1.5 These questions tested a number of basic database skills relating to the organisation of data in a relational database. These questions were relatively easy and were well answered by the majority of the candidates.

Queries:

- 1.6.3. This question required the use of two query criteria and was well answered by most of the candidates.
- 1.6.4. This question tested the ability to use calculations and criteria together, and was generally well answered.
- 1.6.5. This question tested more advanced skills in queries, requiring grouping of data and aggregate functions to count the number of occurrences in a group. This question was generally well answered. However, some candidate lost marks as they did not know how to add aggregate functions to their queries in the Design View query builder.
- 1.6.6. This was an advanced query, where candidates were required to update a table, depending on certain criteria. Weaker candidates struggled to answer this question.

Question 2: Advanced Databases

In this question the candidates were required to import data from a spreadsheet and create a relationship between an existing and the imported table.

- 2.1. Importing a table: generally well answered.
- 2.2. Creating a foreign key in an existing table to be used as a link to the imported table: This question was answered fairly well, although it is unfortunate that Microsoft Access allows relationships to be created between fields whose types do not match. A significant number of candidates lost a mark for linking the mismatched.
- 2.3. Most candidates know how to create relationships effectively.
- 2.4. This question required that candidates demonstrate effective use of the foreign key and was fairly well answered.

Queries:

- 2.5. A simple query using an existing relationship (which was supplied in the original database). Most candidates who attempted the databases could answer this question.
- 2.6. This question was the most difficult question in this section and a small number of candidates received full marks here. Teachers should try to emphasize the use of criteria, together with using totals in a query.

SECTION B: JAVA PROGRAMMING

Question 3: Two-dimensional Arrays

General:

It was clear from the answers to this question that, at a significant number of centres, 2-dimensional arrays were not taught. The comments below refer to candidates who did attempt this question:

- 3.1.1 It must be emphasized to candidates that instructions must be followed. In this question some candidates created a 2-dimensional array whose type was not specified in the question. Additionally, some candidates created 1-dimensional, instead of 2-dimensional, arrays
- 3.1.2 This was a problem-solving question requiring logical thinking. Many of the candidates who attempted this question answered it quite well. This question required candidates to populate the rows of a 2-dimensional array with random numbers from a range of 15 to 90. Furthermore, they were required to ensure that each row's numbers were within a

range of 5 of the value in the first element in each row. More candidates would have answer correctly if they had understood

the calculation of random integers numbers within a minimum and maximum range. The normal way of doing it is:

```
(int) (Math.randomO * (maks-min+1)) + min
```

where:

- maks is the biggest number included in the range
- min is the smallest number included in the range.

(The brackets after the casting are important, else you keep on multiplying with 0).

Most candidates knew how to display and transpose a two-dimensional array. However, sorting of the 2-dimensional array was problematic for number of candidates. This was mostly attributable to the lack of understanding that the sorting required a normal bubble sort, executed repeatedly on rows of data (this requires an additional outer loop for the number of rows).

Question 4: Java classes

General:

Most candidates attempted the first part of this question with success. Since it formed the bulk of the paper, it is extremely concerning that a number of centres' candidates did not even attempt the Java questions. In general, most candidates who attempted this question were able to create a class definition, extend it and populate an array of objects using a text file.

- 4.1. Creating a class definition: One of the main reasons for setting this question was to test correct setting of attributes and methods' accessibility. This is an area that requires some attention, along with the assigning of default values for attributes. It was observed that very few candidates set correct default values for attributes.
- 4.2. This question tested the implementation of inheritance by requiring candidates to extend the class created in 4.1. The correct use of super must be looked at.
- 4.3.1. Most students knew how to count the number of records in a text file, read in data, break down an input String and use it to create an object to be placed into an object array.
- 4.3.2. This question required sorting of an object array according to a specific attribute (course length) and then executing a search on the sorted array. On the whole, the sorting was done correctly, but a significant number of candidates did not go on to complete the searching task.
- 4.3.3. This question was a fairly difficult one and few candidates knew how to create and use typed methods (many candidates, instead, opted to create void methods).
 - The string manipulation task of finding and removing vowels from a string was problematic for some.
 - String methods, like remove() and removeAll(), should be looked at. Many students' programs were not robust enough to handle both
 - Upper case- and lower case letters in the string manipulation task.
 - The general feeling was that the candidates found character and string handling problematic.
- 4.3.4. Most of the candidates attempted this question, managing, at the least, to read user input from the keyboard. However, the following require more attention:
 - Creating an array of mixed objects (i.e. objects of an inheritance hierarchy), searching for a specific object in an array, and printing out user-friendly error messages if the object is not found.
 - The general validation of input should receive greater emphasis so that candidates are encouraged to create robust programs that handle errors well.