The INFT practical paper was written on the first Friday of the fourth term, which placed candidates at a disadvantage because of the reduced amount of revision time available. The exam was also written before many centres had briefed their candidates. Also, for many candidates, it was within a month of writing their preliminary practical exam.

The question paper followed the same format as that of the previous year and most candidates were able to complete the paper within the time allocated.

Teachers need to note that the use of the Java IDE, Ready to Program, is strongly discouraged and that it should be replaced by a more appropriate IDE which makes use of the latest Java Development Kit. Examples of these are JGrasp, JCreator, Eclipse and Netbeans. Candidates should also be discouraged from using the StringTokenizer class since it has been criticized and will be phased out of the Java compiler in the near future. .

Question 1 (SQL)

This question assessed the SQL knowledge of candidates. The SQL code was grouped together in testBandB.java. A few of the weaker candidates found this confusing, but most candidates found it beneficial to separate the SQL code from the output code. This year no marks were deducted for SQL statements that were not inserted at the correct point, but it should not be assumed that this will be the practice in future papers. Please take note that although the queries were not based on related tables, they can still be expected in future papers as multi-table databases are included in the exam guidelines.

Question 1.2 clearly states that the heading of the calculated field must be [Total Due]. However, the result set required it to be TotalDue (one word). Both alternative aliases were accepted.

In Question 1.4, the result set did not work if the field SellingPrice was included in the SQL statement, as requested in the question. No marks were deducted for not including the field SellingPrice.

Although the aim of the note in Question 1.6 was to guide candidates to set the SA field to *false*, no marks were allocated for it, since it did not clearly form part of the given data.

Candidates who copy the SQL code from a database must pay attention to the following:

- Remove the semicolon at the end of the copied code.
- Insert spaces between copied multiple lines that are reduced to one line.

The following example is incorrect:
"SELECT * FROM tblClientsORDER BY tblClients.Surname, tblClients.FName;"
It should be
"SELECT * FROM tblClients ORDER BY tblClients.Surname, tblClients.FName"

Question 2 (Object-orientated approach)

Question 2 tested the object-orientated skills of the candidates. Most candidates attempted this question, but some struggled with it, possibly because their teachers did not cover object orientated concepts sufficiently.

Most candidates realised that cost should be of datatype double (and handled it appropriately), and also that the profit in the method calculatePrice() should be obtained by calling calculateProfit(). However, some candidates used an attribute profit, which is only acceptable if the method calculateProfit() has been called in the constructor. Many candidates added an additional attribute for name because the field was present in the text file *Extras.txt.* Teachers must ensure that their candidates follow the instructions for the object attributes, as marks may be deducted for incorrect attributes.

Candidates need to pay attention to the example output given when designing the toString(). Three marks were allocated for having output that corresponds with the example output. Confusion was caused by the instruction "…formatted as

follows: Item type <tab> Cost<tab>Profit<tab>Final Price". Candidates put labels in the string of the toString(), which resulted in them losing one mark in the driver class for not having the following heading:
"Item type Cost Profit Price".

The instruction, "*display an appropriate message to say that there are no extra charges for this guest, if no items have been found*", implies that an if/else or if/if structure needs to be used. Candidates also lost marks for not putting this code outside the loop structure. Most of the candidates tested for the existence of text file by means of if()/else or try/catch statements, but forgot to close()the file reader and/or exit() the program.

Question 3 (Problem solving)

String manipulation was at the heart of this question. Candidates with a sound foundation in Grade 10 and 11 successfully completed the program. The note: "*Use modular programming techniques when solving this problem*" was not taken into consideration in the mark allocation. Weaker candidates were confused by the instruction to overwrite the old phone numbers with the new ones. This was further complicated by the fact that a menu was given, which means that the array will be replaced when Option A is selected. Candidates were not penalised for not replacing the original array. Question 3.2 required the inner loop to break as soon as a duplicate is found. Very few candidates made provision for this. Since this question is orientated towards problem solving, it may contain two dimensional arrays in the future, as per the exam guidelines.