May 2012 subject reports

COMPUTER SCIENCE

Overall grade boundaries

Higher level

Grade:	1	2	3	4	5	6	7
Mark range:	0-12	13-25	26-34	35-45	46-56	57-67	68-100

Standard level

Grade:	1	2	3	4	5	6	7
Mark range:	0-12	13-25	26-36	37-46	47-57	58-68	69-100

Higher level program dossier

Component grade boundaries

Grade:	1	2	3	4	5	6	7
Mark range:	0-3	4-7	8-12	13-17	18-23	24-28	29-35

The range and suitability of the work submitted

The work submitted showed a steady improvement in applying the mastery factors correctly and fewer candidates were penalised significantly. It would be helpful if candidates and teachers annotated the specific code section related to a mastery claim as often page numbering can be incorrect because of changes to listing etc. which can make it difficult to verify.

Most HL candidates will gain mastery of the SL factors, but it is still the case that this is sometimes not claimed.

Again, any HL candidate that uses a GUI will satisfy the related mastery factor. Teachers should check these two mastery factors to ensure they have been claimed.

The problems selected were nearly all appropriate, these mostly focused on a database. One area for teachers to consider is to encourage candidates to use an index into the database. To access data in a random access file the position needs to be known, this can either be found using a hashing method or by doing a linear search, alternatively a key can be used that is paired with the position number. This data can be stored in a binary tree and allow candidates to claim mastery of an ADT and potentially the OOPS mastery factors as well.

Nearly all candidates presented the dossier well with clear sections and good use of sub-headings. One area that prevented many candidates gaining full marks for the report (D2) was the lack of comment on alternative design processes. Otherwise the report showed candidates had a good awareness of the importance of reflection and review.

Candidate performance against each criterion

A1 Analysis - this was well done with most candidates showing data collection processes and clear discussion of the problem

A2 Objectives - most candidate did this well producing a clear set of objectives, one area to look at is making sure the objectives are clearly related to the analysis.

A3 Prototype - on the whole done well. Some candidates did not include an initial design. To gain full marks it is important to show feedback from the user.

B1 Data Structures - continues to be an area of weakness. Candidates need to outline, justify and illustrate, using real data, the operations on the data structures. Generic diagrams are not sufficient, it is expected that the discussion is directly related to the specifics of the problem being solved. Both memory data structures eg arrays, linked lists, binary trees and disk structure like random access files and sequential access files need to be addressed, along with class structures used.

B2 Algorithms - done reasonably well, although a number of candidates are just presenting computer code or not providing sufficient detail. All major algorithms should be included.

B3 Modular design - most candidates presented some form of diagram which showed links but many did not explain what the modules did or how the links operated.

C1 Program layout on the whole was well done, it is expected that there are header and inline comments, the layout shows consistent indentation and that meaningful variable names are used.

C2 Error handling - nearly all candidates trapped some I/O errors, but to gain full marks the candidate needs to consider most possibilities and provide evidence.

C3 Success of the program - most did this well by using a table listing the objectives, a conclusion and listing reference. Some candidates did not have a separate section and this is required.

D1 Hardcopies - mostly done well, not always clear if the list was complete and it is advisable to link to C3 and A2 to state clearly that all objectives have been tested.

D2 Report - done well on the whole with the use of sub-headings. However, as stated above, many did not review their design approach or alternative approaches. This needs to be done for full marks.

Recommendations for the teaching of future candidates

Mastery Factors

Candidates claiming master related to random access files must use a RAF, some candidates are still using text files.

File search is possible with either text or random access files.

Inheritance can't be claimed for extending a basic Swing component e.g. Frame or Panel.

Polymorphism is generally questionable for a default and one other constructor, be careful with this claim.

Encapsulation requires the data variables to be hidden as private.

Hierarchical Composite Data Type - a composite data type is a simple record structure i.e. like a Structure in C/C++. A HCDS is generally a list of records.

Parsing can't be claimed for standard Java methods to convert text to a number or reading from a file using the end of line marker. It is expected that some delimeter is used with StringTokenizer or Split, or the candidate directing coding the splitting.

To claim a fully detailed ADT requires a full abstract description of all possible operations and error checks. Most ADT claimed by candidates combine problem specific processing and some abstract operations e.g. add, find, delete, length, isEmpty - this is acceptable.

The adding and deleting records from a text file must be done by NOT reading the data into a memory data structure like an array, a new text file must be created and the data from the old file read record by record to locate the addition or deletion point and the files must be ORDERED!

Linking the Exam to the Dossier

In the current course the HL dossier allows direct linking to concepts examined in the HL algorithms: arrays, text and RAFs; Object Oriented Concepts of inheritance, polymorphism and encapsulation; data structures such as: linked lists, stacks, queues and binary trees, and sorting and searching. Clearly, a dossier will normally only use a sub-set of these concepts but it is important to reinforce the learning through practical application.

Structure of the write-up

Section C3 MUST be included, it provides a way to show that ALL the objectives have been meet and hardcopy is linked to each. If this is not done it is often difficult to determine the correctness of the mark awarded. The report (D2) should include a summary of this table in the discussion of effectiveness of the solution.

Standard level Program dossier

Component grade boundaries

Grade:	1	2	3	4	5	6	7
Mark range:	0 - 4	5 - 9	10 - 14	15 - 19	20 - 24	25 - 29	30 - 35

The range and suitability of the work submitted

Almost all projects matched the general criteria for Computer Science projects with regards to both the programming language (Java) and the required documentation. There was one notable exception where a project was written in php, which resulted in the project being penalised.

Candidate performance against each criterion

Section A – Analysis

The main weakness here (as always) is the attempt by many candidates to complete the project without the serious participation of an end-user. This inevitably leads to A1 dealing more with the solution than the actual problem. Objectives (in A2) can´t match up with the user´s goals in A1 (because there isn´t a user) and any feedback to the prototype in A3 will have been added by the candidate himself. A serious analysis can only be made when the candidate has to face a client with a previously unknown problem.

Prototyping has improved throughout the life of this specification with the inclusion of screen-shots of the various menus / sub-menus, although not always preceded by an initial design. To gain credit, feedback must be both substantial and credible.

Section B – Design

It is appreciated that each set of candidates sees these criteria for the first (and only) time. However, this is not the case with the majority of teachers, many of whom fail to clarify how section B1 should be approached. The data structures that must be discussed (with examples of actual data) at SL are files, arrays and objects as data records. It is surprising how many candidates fail to demonstrate their understanding of the difference between structures used for permanent storage and those used temporarily during the running of the program.

Algorithms (B2) still tend to be, for many, a reluctant afterthought. Algorithms that are basically final code will not receive credit. Modular organisation should diagrammatically show the connections between various modules / classes and be accompanied with sufficiently detailed descriptions.

Section C – Development

It is understandable that candidates would want to include GUI interfaces (C1), but the code generated automatically in this respect should be highlighted in some way – this code cannot be used to claim for mastery. Teachers must take responsibility for insisting that all code written is thoroughly documented (as it is written). Claims for mastery should be clearly highlighted in the code. The use of incorrect page numbers means that it is no always easy to find the code relating to these claims.

Error-handling (C2) is always handled better when accompanied with screen-shots and code snippets. Success of the program (C3) should link initial objectives (in A2) with the appropriate test runs.

Section D – Documentation

The main weakness in the testing (D1) is the failure to show the results of any changes in the data. For example, if a record is modified in some way, or records are sorted, then the records should be listed both before and after the operation in question. If not, how can the success of this test run be proven? Large data sets must be originally input, so as to demonstrate serious testing (the original text files could simply be added through use of a notepad or similar programme).

For the awarding of mastery aspects, runs conclusively demonstrating their success must be shown. Otherwise, aspects such as file I/O, sorting and searching will not be awarded. Note that for file I/O, both input and output must be explicitly shown.

The final evaluation (D2) rarely gets the treatment it deserves, and is also one of the criteria in which teachers over-mark the most. The actual running of the program should first be outlined, followed by a serious look at efficiency, effectively, possible future improvement and alternative designs (the latter referring to significant re-designs).

Recommendations for the teaching of future candidates

- get a real end-user and keep the person seriously involved test with large data-sets and show clear evidence that all modifications have been made

- dedicate appropriate time to the final evaluation

Higher level paper one

Component grade boundaries

Grade:	1	2	3	4	5	6	7
Mark range:	0-14	15-28	29-35	36-43	44-51	52-59	60-100

The areas of the programme that proved difficult for candidates

Most candidates performed reasonably well. Many candidates have difficulties to respond to questions which requires the application of the knowledge instead of a straightforward "text book" responses.

The areas of the programme in which candidates appeared well prepared

There is a wide knowledge base of candidates appearing for this examination. There were only a few candidates who were very poor in their performance. The majority showed a good coverage of the course.

The strengths and weaknesses of candidates in the treatment of individual questions

SECTION A

Question 1

Most candidates stated a purpose of a compiler instead of a way in which it helps in the development of a program.

Question 2

Most candidates answered this question well.

Question 3

Many candidates described examples of verification and validation which is not the answer to this question.

A few answered the question correctly. They explained that validation prevents the propagation of invalid data in the system by discarding data not in prescribed format for processing. Verification aims to establish the accuracy of data and limits errors introduced by human operators in inputting data.

Question 4

Below average marks earned. Many candidates were not able to outline the resource monitoring function of an operating system.

Question 5

Part (a) was generally well answered. Many candidates failed to answer correctly Part (b).

Question 6

Some candidates confused formatting with fragmentation.

Question 7

Well answered question.

Question 8

A surprising number of candidates incorrectly calculated the number of different colours that could be represented but were able to calculate the size of the graphic file.

Question 9

Well answered question.

Question 10

Many candidates answered well this question. Most candidates correctly stated the Boolean expression that corresponds to the given circuit without simplification. Some candidates were unable to provide a simplified Boolean expression.

Question 11

Most candidates described the difference between serial and parallel transmission but did not identify the advantages of serial over parallel transmission.

Question 12

Well answered question.

Question 13

Well answered question.

SECTION B

Question 14

This question involving system analysis and design was answered relatively well by majority of candidates. Many candidates did not use the appropriate symbols to construct the system flowchart.

Question 15

Most candidates who attempted this question did well. Only a few candidates were not able to trace the recursive method. Candidates tended to have either have good programming skills and understanding or not.

Question 16

A reasonable number of candidates on the whole knew what polling means and how a CPU handles an interrupt. In Part (d) most candidates failed to achieve full marks as result of incomplete and vague answers.

Question 17

Many candidates devoted a great deal of space and time to explain how direct access differs from sequential access but never explained why direct access is appropriate for accessing data in this situation and how a record can be accessed quickly if the file does not consist of fixed-length records.

Question 18

Similarly to Question 15, most candidates who attempted this question did either very well or very poorly. In Part (d) some candidates wrote answers that were too vague and of course.

Question 19

Most candidates provide a labelled diagram showing a suitable hybrid network. Selection of physical transmission media seemed to confuse some candidates. Some candidates wrote explanations in Part (c) in vague and non specific ways thus received rather few marks.

Recommendations for the teaching of future candidates

Exam practice- many candidates could have earned more marks by being less subjective and general.

Questions should be analyzed in order to determine what is being asked. All answers should contain computer science and not general observations.

Higher level paper two

Component grade boundaries

Grade:	1	2	3	4	5	6	7
Mark range:	0-13	14-27	28-32	33-42	43-52	53-62	63-100

The areas of the programme and examination which appeared difficult for the candidates

“Files”, as always, proves a difficult topic. The Case Study was not as well attempted as in previous years (see Qu.4 below).

The areas of the programme and examination in which candidates appeared well prepared

Algorithms dealing with processing 2D arrays and the movement through linked lists.

The strengths and weaknesses of the candidates in the treatment of individual questions

Question 1: Algorithms / 2D Arrays

Candidates are by now well-versed in dealing with 2D arrays and the various search based algorithms associated with them. Most constructed the circle method (1b) competently, although some glossed over the opening paragraph and decided there were only 4 towns.

1(c) proved more difficult, not so much in the syntax but in the logic required. Many chose to solve the problem in two separate steps which was incorrect. However, these candidates were still able to gain some marks from fulfilling the various conditions.

The wording in 1(d) was important. HL candidates cannot expect to receive credit for descriptions that are vague in nature. They needed to specify a array of towns that had positions which corresponded to the positions of the original distance array.

1(e) tended to differentiate between the candidates with the more able candidates selecting appropriate structures.

Question 2: Linked Lists / Objects

Similarly with arrays, it was clear that a lot of work had been put into dealing with linked lists, with many candidates comfortably handling movement through a list. Having an array of linked lists was probably a new concept for the candidates, but it did seem to prove to be a problem.

Both parts of 2(e) proved difficult. The better candidates realized that a linear search would ultimately prove untenable in part (i) and suggested various improvements in part (ii), the majority increasing the number of letters used thus creating more positions.

Correctly reassigning the head in 2(c) led to some confusion, with some losing access to the relevant list.

For both Questions 1 and 2 there always seem to be a small but significant group of candidates who cannot deal with algorithms at this level, and who should, perhaps, have been entered at SL

Question 3: Files

This was a reasonable straight-forward question with some SL parts. Candidates who could handle dynamic structure algorithms in Qu. 2 could not necessarily apply the theory to actual applications as in Qu. 3

The top HL candidates knew binary trees well enough to answer parts (b) and (e) but few gained full marks for the basic updating of stock records in 2c(ii), Surprising few chose to link the dynamic structure directly to the correct record in the file, and some chose incorrectly to import the stock level fields into the dynamic structure.

Question 4 Case Study

It was clear that in spite of the fact that 40% of the marks from this paper comes from this question, a corresponding amount of time is not necessarily being devoted to the Case Study in the teaching programme. Teachers who pass the responsibility for its study onto the candidates at the end of the course are dong them a disservice. The introduction at the beginning of the present Case Study made it clear that significant research was expected into the various topics, but it was equally clear that for many candidates this did not happen, and that many questions were answered more from general knowledge

Consequently the marks for this question were lower than in previous years.

However, some of the answers were quite inventive but likely to have been related to their own personal experiences.

Recommendations for the teaching of future candidates

The strongest recommendation deals with the preparation for the Case Study. This must be a teacher-led investigation with sufficient time dedicated for exploring the various topics to a reasonable depth.

Also, relating dynamic structures to actual applications is important in order to put the theory in context.

Standard level paper one

Component grade boundaries

Grade:	1	2	3	4	5	6	7
Mark range:	0-7	8-15	16-21	22-28	29-35	36-42	43-70

The areas of the programme and examination which appeared difficult for the candidates

There was some difficulty in tracing and examining in depth the algorithm. Specific parts of the syllabus such as MICR, the details of disk formatting, resource monitoring and the specifications of the CPU seemed unfamiliar to many candidates.

Generally there were many weak scripts showing very little application and analysis.

The levels of knowledge, understanding and skill demonstrated

The elements of system life cycle were familiar to most candidates who were able to discuss data collection, data entry and test data. Similarly hexadecimal, binary and decimal representation of numbers was an area in which they seemed well prepared. It was pleasing to see that the two diagrams required were clearly drawn and labelled for the most part, although there were significant gaps in the system flowchart.

The strengths and weaknesses of candidates in the treatment of individual questions

Section A

Question 1 was generally well answered showing that features if compilation were well know. Some failed to combine this knowledge to the development of a program.

Question 2 was answered well by most candidates demonstrating full knowledge on the features and uses of verification and validation.

The answers to question 3 were disappointing, demonstrating little knowledge of the components of a CPU in a microprocessor. Many assumed that no ALU was present.

Question 4. There were quite a few guesses or question repeats for resource monitoring, which was evidently an unknown phrase to many of the candidates.

Question 5. Types of data necessary for testing were generally identified.

Question 6. The phrase “for its first use by an operating system” confused many who proceeded to explain how the operating system could be stored on the disk, instead of outlying the function of formatting.

Question 7. A surprising number of candidates had no idea of the number of colours that could be represented. On the other hand, others spent a great deal of time doing the correct calculation rather than answering 2^{15}. However, the size of the graphics file was mostly correctly given whilst the suggested methods of storing the graphics file in 12 kB RAM were weak.

Question 8. The output to the algorithm was almost always correctly given.

Question 9. Decimal and hexadecimal answers were generally correct.

Question 10. The role of cache was correctly outlined by many.

Question 11. Very few could identify an advantage of MICR over OCR.

Question 12. The disadvantages of interviews were generally well explained.

Section B

Question 13. The diagrams of a hybrid network in part a) were often clear, labelled and correct. The one element that was almost always absent was a switch or hub to connect the networks. There were no particular problems with the rest of the question which was often the best one in section B for the candidates.

Question 14. This question was the most difficult for many candidates. In part a) some candidates repeated the while condition rather than the terminating condition. The contents of *array b* were fairly straightforward to trace for candidates but *sum* proved more difficult often displaying an inability to add 13 and 9. The rest of the question was completed correctly by only a handful of the candidates. Improbable answers such as entering negative numbers were proposed.

Question 15. The weakest section of this question was the system flowchart, which often lacked the dispensing of the fuel and the calculation of the bill. The description of the process required in part a) and the extra information needed to send in part b) were generally as correctly answered as was part d).

Question 16. Parts b) and d) proved difficult for candidates. Despite the information given about the file, few candidates could go further than say that the file would get bigger and take longer to search if another item was added. The rest of the question was reasonably well answered but there was evidence that some candidates were rushed at the end.

Recommendations for the teaching of future candidates

Ensure that candidates are secure in the functioning of hardware components and that they have practise in answering extended questions which require the application of knowledge. In particular, that they read carefully such questions before answering. Quizzes can help to reinforce basic facts and application and analysis can be developed by exploring the use of computer hardware and software in different scenarios

Standard level paper two

Component grade boundaries

Grade:	1	2	3	4	5	6	7
Mark range:	0-10	11-20	21-26	27-31	32-37	38-42	43-70

The areas of the programme and examination which appeared difficult for the candidates

The concept of 'scope of a variable', Q(a)(i), was not well understood. Variables declared within a block of code can only be addressed within that block; this can be within an if statement or for loop.

```
for (int c=0; c

{
   c is local to the code block { } of the for loop.
}
```

The concept of an array being used as a counter for a sequence of related values was not well understood. An array can be indexed and incremented rather than using a set of separate counters e.g c1, c2, c3, c4 etc. Such a sequence should suggest a list data structure, an array, be used.

Candidates showed a reasonable understanding and ability to construct algorithms. Loops were generally correct and the overall structure of the statements showed good layout and logical sequence. A number of candidates were able to make use of a pre-written class and call a use data returned by methods of this class.

Candidates showed a reasonable understanding of the concepts of a sequential access file and how such a file might be used in a business - see Q2 - by using the barcode as a unique identifier, and where able to construct a reasonably complex algorithm related to business processing.

Some candidates showed a sound understanding of the case study giving answers that drew directly on the specifics of the case study.

The levels of knowledge, understanding and skills demonstrated

The concept of the use of the identifier private was well understood.

Candidates were able to perform traces on algorithm in Q1 (a)(iii) & Q2 (c), which was pleasing as this is an important skill.

Whilst candidates showed a reasonable ability to construct algorithms many did not show a good understanding of the use of predefined classes and methods. For example, passing parameters and assigning returned values.

A number of candidates confused the location of a file i.e. on the server, and not as part of the barcode: the barcode provides a number only that can be used to identify the data in the file.

In general the case candidate was not as well answered as previous years. It is very important that candidates use the details of the case study in their answers, it is also important to avoid vague general answers.

The strengths and weaknesses of candidates in the treatment of individual questions

Q1 (a)(i) poorly answered, see comment above.

Q1 (a(ii) well answered

Q1 (iii) generally well answered, candidates need to make sure they show the output in correct order.

Q1 (b)(i) & (ii) well answered.

Q1 (c) not well answered, many candidates listed a single variable by miss-reading the question. The indention was for an array to be used - see the above comment.

Q1 (d) generally well done. Most candidates were able to pass the data, define the required loops, select the required word and then the required letter and outline a test and increment a counter.

Q2 (a) well answered, although candidates need to understand that a barcode represents a number nothing else!

Q2 (b) well answered.

Q2 (c) the first to class run are always the main and then constructor. Not that well answered.

Q2 (d) reasonably well answered. Many candidates made good use of the Customer class, the methods in the class and used an array. The loops generally were included correctly and the basic logic was sound across a range of responses. The weakness, as outlined above, related to the calling, passing data and assigning returned values.

Q2 (e) reasonably well answered.

Q2 (f) candidates provided a range of workable answers.

Q3 (a) Candidates often did not make good use of the material in the case study by giving vague answers that did not seem to refer to the case study.

Q3 (b) Again candidates often did not seem to relate their answers to the specifics of the case study.

Q3 (c) reasonably answered.

Q3 (d) reasonably answered.

Q3 (e)(i) candidates appreciated the importance of security but the vast number did not make good use of the case study to answer the question.

Q3 (e)(ii) Many provided examples, but a number of candidates gave general answers that did not relate to the case study.

Q3 (e)(iii) Reasonably answered, however, it was clear some candidates were not familiar with the case study.

Q3 (f) adequately answered, many highlighted advantages and disadvantages, but often did not seem to highlight any technical issues contributing to these.

Recommendations for the teaching of future candidates

Teachers need to ensure that candidates:

- understand the concept of variable scope;
- are able to use arrays in a variety of contexts;
- understand the use of sequential files and how data can be processed using files and are able to use a provided Class to construct an algorithm and use methods within the Class to call methods.

The case study needs to be better used by candidates to support their answers.