International Baccalaureate®
Baccalauréat International
Bachillerato Internacional

COMPUTER SCIENCE

Overall grade boundaries

Higher level

Grade:	1	2	3	4	5	6	7
Mark range:	0 – 13	14 – 27	28 – 37	38 – 49	50 – 61	62 – 73	74 – 100

Standard level

Grade:	1	2	3	4	5	6	7
Mark range:	0 – 14	15 – 28	29 – 42	43 – 53	54 – 64	65 – 75	76 – 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 project topics chosen were wide and varied, with many *“real world”* applications being developed. A few were simplistic, in that only a few pieces of date were chosen to be recorded for a particular project, but this had more to do with the coder’s decision, as opposed to the topic itself.

The work ranged from excellent to very poor. Most dossiers addressed a suitable problem and wrote computer programs that ran successfully. However, some solutions were very weak and did not provide a usable solution. In a few cases the documentation was lacking.

Students in several schools used the criteria from the previous years and hence included Usability and User Documentation sections. Although this only wasted time some students used the Mastery Factors list from previous years and hence failed to claim mastery of aspects like 2D arrays.

That was surprising, as the teachers used the correct cover sheets containing the correct mastery factors.

In most cases, students in the same school had the same strengths and weaknesses. To summarize:

- Stage A was often well done
- Stage B mediocre, often poor
- Stage C some programs much too long while at the other end of the scale some programs never ran successfully or were never demonstrated to do so.
- Stage D most dossiers contained too little sample output, or failed to test all the features

Some students entered at this level produced program code that would have been better suited to a Standard Level dossier.

Candidate performance against each criterion

Criterion A1

Generally a good section, with end user interviews, discussion of previous solutions, and the development of data flow diagrams. Some projects went into far too much detail in reviewing previous solutions (12 pages) but these were the exception and not the rule. Some candidates tried to fit all the documentation into one page.

Some students analyzed problems and included sample input and output data, user requests and background information. However all of these were only found in a few dossiers.

Criterion A2

Some students stated only very vague goals (e.g. should function well) and had trouble later defining the solution clearly. Continuing on from the above, some dossiers had more discussion in this section to justify their goals than found in A1. Other trends included the listing of obvious or vague goals, such as *“create a friendly interface”*.

Criterion A3

Evidence of user feedback is often patchy or unconvincing. Some students continue to misunderstand the requirement for an initial design, which can be a simple module diagram arising naturally out analysis of user needs as described in sections A1 and A2.

Criterion B1

As a whole, section B continues to be the most volatile part in terms of range of marks. Writing this section after coding was complete was evident in some cases, even to the point of using the wrong tense; *“my project has the following classes”*, as opposed to *“my project will utilize the following data structure for these three reasons”*. Some had excellent diagrams for ADT operations, others just copied in classes from code.

Many students chose inappropriate data-structures just to satisfy the Mastery Factor requirements, even when it would have been possible to choose a better data-structure and still get mastery marks.

Criterion B2

Even though the requirement has been considerably reduced, many students still simply copy code or algorithm fragments from the solution and then describe them.

An outline of major algorithms in some kind of pseudo-code or other text form is required for this section; this should be done at the design stage, before any code has been written.

Criterion B3

There seems to be no clear concept of what's required here - lots of different approaches were used. A simple module diagram outlining the proposed solution is all that is required. Pages of description of the classes in the code listing are not required and contribute very little that comments in the code would not.

Candidates may depart from their Section B designs during the programming of the solution.

Criterion C1

Code listings are generally good. Candidates still do not always mark out the code that has been generated by an IDE from code they have written themselves.

Criterion C2

This section should refer to **Error Handling** only. **Usability** is not required.

Generally well done, some candidates did not use code snippets or specific page references many included a reasonably full range of error detection methods.

Criterion C3

There is no separate documentation required for this section. However, it is not a bad idea for candidates to refer to section D1 to indicate where their solution meets the criteria set out in section A2. A simple table could be used.

DOCUMENTATION

Criterion D1

As in all years past, almost all students provided too little sample output. In some cases students only tested error cases and did not show the program functioning successfully with normal data. Considering that these were mostly database oriented programs, it is strange that so few students ever showed a list of data - say 5 people's names with associated fields.

Also in many solutions, additions, edits and deletes from lists (either files or ADT's) were never explicitly shown. This means that the examiner could try (but is not obliged) to determine from the code listings, whether these functions actually worked or not. This is a very big risk for candidates to take given the reduction in marks that can occur from having one or more mastery aspects discounted.

About two thirds of the sample runs (and there MUST be more than one) should show the program operating normally and as claimed.

Criterion D2

As for Standard Level, this section is not always thoroughly done. Efficiency should be discussed at this level and suggestions for future modifications should be sensible and realistic – ie within the capabilities of the proposer.

User Instructions are no longer required.

Recommendations for the Teaching of Future Candidates

Candidates are advised to follow the format and guidelines for dossier specified in the Programme Guide 2010. Some schools are evidently still using the older guide.

Students should not write the program before doing the analysis in Stage A and/or the design in Stage B. Such an approach wastes time and effort and often produces a faulty or severely limited result.

Some teachers have either misunderstood or misinterpreted some Mastery Factors. Teachers should clarify their own understanding and communicate this to their students BEFORE the students start writing the program. It is also an excellent idea to get advice from the Online Curriculum Centre when in doubt about what constitutes mastery.

The most poorly understood mastery aspects are:

- Data is often simply **appended** to an RAF instead of added
- Recursion is often trivial or incorrectly implemented
- Polymorphism is usually trivial if just multiple constructors are referred to
- Encapsulation is ineffective if instance variables are not explicitly made private
- Parsing a text file requires more than the simple application of Java library methods
- An ADT must be designed and written by the candidate, and described effectively in section B1
- Using text files will require a new file to be produced when inserting or deleting – the list of items cannot be held in RAM

The problem should be sufficiently limited that it is possible to produce a meaningful solution in the time available. Goals should be clear and precise, and the resulting programs should provide a usable (not necessarily perfect) solution for the problem.

Aside from what was note above, it seemed that more students are taking advantage of classes and utilities in the Java language in order to create efficient and productive applications. This is a good sign, and **indicates** that one of the strengths of Java is being properly leveraged.

While candidates are encouraged to use and adapt code from the internet or other sources if code is copied, no credit for mastery aspects can be awarded and if derived a reference to the original is essential to avoid suspicion of plagiarism which carries severe penalties in almost all cases. The use and adaptation of such code should be explicitly discussed in section B at the design stage as this will allow the teacher to advise the candidate properly.

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

The selection of problems and the presentation of dossiers in this session were generally good or very good. As ever, some of the problems attempted were overly complex for a dossier at Standard Level. Candidates at Standard Level should be deterred from attempting problems which would require the complexity of a relational database for example. They should be advised to limit themselves to one major record-style class with a few fields of different types where it is reasonable to do so. To attempt more is unduly increasing the workload of candidates which may well impact their study of other subjects as well.

Students who chose a suitable end user (such as a teacher in their school or close relative) found it much easier to gather the required data for the analysis and goal-setting parts. This was usually because the ability to talk to the end user gave a much deeper understanding of the problem, leading to a well-informed analysis and a more thorough design. By contrast, students who reverse-engineered the analysis and design after having written the program did not score well in these sections.

Teachers and candidates need to be sure that they both understand clearly what each of the mastery aspects entail and should ask for advice from, for example, the Online Curriculum Centre, if necessary. Simply guessing is undesirable since the mastery factor can have a very large effect on the final mark.

Where simple "setter" and "getter" methods and constructors are used as part of Class the same methods should not also be used to claim mastery of methods with parameters and methods with return values if only because such methods are usually trivial. If this approach is desirable then candidates should consider adding further code in the methods, for example to test (validate) the value of the parameter. In this case, the instance variables need to be explicitly declared as private.

Candidates MUST provide sample output of normal runs of their programs – this is essential for the moderator to confirm that mastery of aspects has been achieved, the program actually works as stated and to confirm the teacher’s assessment in C1. The testing of abnormal and extreme data is less important.

It is virtually impossible to confirm mastery of certain aspects, such as arrays, file i/o sorting and searching without hard copy evidence. The only recourse a moderator has is to try and interpret the potential success or otherwise from the code listing and few moderators have this kind of patience (nor are they expected to). It is the candidate's responsibility to demonstrate success.

Candidates are encouraged to use and adapt code from the internet or other sources. However, if copied, no credit for mastery aspects can be awarded and if derived a reference to the original is essential to avoid suspicion of plagiarism which carries severe penalties in almost all cases.

Collaborative work is forbidden.

Candidate performance against each criterion

ANALYSIS

Criterion A1

Most students at this level make the unfortunate error of assuming that the moderator already knows what their problem is. They should be encouraged to give background information, use diagrams and pictures and to give sample data from the problem domain.

There should be evidence of data-collection. Many candidates did not provide genuine evidence that data had been collected. Evidence refers to physical evidence like screenshots of other solutions, membership forms, log books etc. If an interview has taken place there should be a transcript and it should be non-trivial - this means that actual data should be presented - actual numbers and strings from the problem domain. Many dossiers present explanations - e.g. "name and phone number" - but there should also be sample data presented - e.g. "Fred Thomas, 312-4567". Students who were able to provide this data almost always scored better overall on their dossier than those whose user was illusory or patently made up.

In an ideal case, a small manual system is being improved and existing documents from this system can be invaluable in providing sample data. Students of computer science should be adept at using scanners and digital cameras to capture and include relevant data – most other students are by now.

Criterion A2

Criteria need to be specific and measurable to have any real meaning. *"The program should be efficient and user friendly"* is too general and not measurable. Criteria for efficiency and user friendliness would need to be included in order to count towards marks in this section.

"The user interface will have consistent placement of command buttons to ease user navigation as shown in the prototype" would be specific and measurable, for example.

Students should explicitly relate the goals to the analysis. This implies some sort of explanation why each goal is important. For example:

> *"The list of cars should be searchable by colour or number of doors as users often have specific requirements when purchasing a used car"*

This section is better done in numbered points rather than written in essay format which is both hard to read and to refer back to in later sections.

Criterion A3

Many dossiers did not include a design or user feedback.

This is such a useful, practical and simple exercise to carry out that it is sad that better attempts are not made at it. A good prototype for a real user really lets the student get a good feel for the problem as an information problem, not just a computer programming exercise.

The initial design can be a very basic data-flow diagram or outline. The prototype should be something appropriate for discussion with the end user - preferably a user interface, but should also include sample data, not just menu items.

The discussion with the end user should be documented in some way; this was not always done or was rather trivial such as; "*The user liked what he saw"*. Some students simply include their screenshots from the solution here; prototyping is a useful transferable skill so this is a lost opportunity sadly.

DESIGN

The design section continues to cause students the most problems it seems and this is usually because a thorough analysis of the problem has not been completed.

Criterion B1

This presents an excellent teaching opportunity if candidates share their thoughts on data structures and are encouraged to explain them to each other.

A data structure at Standard Level is usually a record, an array, a text file and several primitives of different types will often be used to store data for the solution. A brief explanation of these, perhaps a justification for their use, sample data and a diagram would satisfy the requirements of this section and give the candidates some insight into algorithm development as well.

Diagrams showing how the contents of arrays and files would change during program execution are ideal – but rarely done by candidates.

Criterion B2

Some candidates are still giving pre-conditions, post-conditions for algorithms although these are no longer required. Java code should not be copied into and/or adapted and described for this section. An outline of major algorithms in some kind of pseudo-code or other text form is all that is required. This should, of course, be done at the design stage, before any code has been written.

Criterion B3

An outline of classes already written in Java code is not suitable for this section.

A relatively simple set of modules, constructed during the design stage, making clear the connections to algorithms and data structures will be sufficient for most candidates. If in doubt, a diagram is a simpler and better approach for most candidates as compared to a text-based one.

Candidates may depart from their Section B designs during the programming of the solution.

THE PROGRAM

Criterion C1

This should not be difficult to do with a modern IDE and most candidates have done well but there have been some truly shocking examples of code presentation in this session.

A mono-spaced font, proper and consistent indentation, meaningful identifier names and comments for each class, method and instance variable should be the minimum that teachers accept. If this is insisted upon before teacher advice is given at all stages of the dossier it will become a good working habit for the student.

Criterion C2

Usability is no longer required. This section should now refer only to Error Handling.

This section should refer to code examples either quoting code or referring to specific methods and lines precisely – either by presenting them again in this section or by reference to C1 Code Listing.

Better candidates provided an explanation of the error trapping routines detailing the possible error to be trapped and how the code detects and prevents it occurring.

Criterion C3

Many teachers provide helpful comments with their dossier submissions and this is immensely useful to the moderator in evaluating the mark awarded in C3.

While it is not a requirement, a good approach is for candidates to provide a description of how the program fulfilled each goal, including reference hard-copy output that shows that the program actually functioned as expected. Reflection on what students have produced is a very useful teaching tool.

This implies that a good set of sample runs (not just one) with normal data have been carried out. It could be more usefully produced after section D1 has been completed.

DOCUMENTATION

Criterion D1

Very few candidates performed a series of strategic test runs instead showing isolated tests of single events.

Candidates need to spend more time following valid data through the normal life of the system than just showing that isolated algorithms work.

Candidates should carefully pick screenshots that demonstrate that each criterion in A2 has been achieved and that all claimed mastery aspects are working. The guide states that one run with valid data is rarely sufficient. More sample runs should be made with valid data than with invalid data. It is much better that the candidate emphasizes what has been achieved. Tables of proposed test data are not essential but candidates who did this approach generally scored better.

Testing of invalid output should not be neglected but probably should be about one third of the total screenshots produced rather than nine tenths.

Evidence that multiple records can be added to arrays and files is essential for the confirmation of the award of these mastery aspects.

Annotations should be added and they should be meaningful and non-trivial in order to earn marks.

Criterion D2

This criterion was not well handled by many candidates. Most outlined the solution and discussed possible improvements and effectiveness.

Often candidates did not understand what efficiency was, some consideration to limiting the number of iterations required for processes to complete, disk space usage or something similar is useful.

A few candidates after suggesting improvements to the program also considered alternative solutions to the problem such as use of a spreadsheet or a paper based method and compared the merits of these systems to that of their program.

Many schools included two or three pages at the end documenting how mastery aspects are achieved referring to the line of code in the program.

Many candidates are justifiably proud of their achievements but this should not be the sole consideration in an evaluation.

Recommendations for the teaching of future candidates

In last year's Subject Report it was stated that *"A good SL dossier for the new 2010 criteria can probably be achieved in 40 pages"*. This remains the case, although there are signs that this has been interpreted too literally. Only a high quality dossier in which every section is carefully written and to the point can do this.

Throwing in anything and everything to reach those 40 pages is not the answer.

Dossiers which were produced around an already complete program did not score well, the better dossiers followed the design process and usually involved a database, allowing full scope for mark earning under all headings.

UML diagrams could help all of section B in general, as well as other graphical tools that lend themselves to the design stage. For example, see http://structorizer.fisch.lu/ for a free, robust algorithm definition tool.

The most important factors in achieving a good dossier mark continue to be:

- having a user, who is known to the candidate,
- with a real problem of suitable scope **and**
- following through the analysis and design stages **before**
- coding the solution.

Higher level paper one

Component grade boundaries

Grade:	1	2	3	4	5	6	7
Mark range:	0 – 15	16 – 30	31 – 40	41 – 50	51 – 60	61 – 70	71 – 100

The areas of the program that proved difficult for candidates

Overall many candidates performed reasonably well. However, one of the weakest areas was basic computer knowledge.

Boolean Logic (Question 15)

Only a few candidates were able to construct a Boolean logic circuit after simplifying the long expression obtained from the truth table.

Recursion (Question 16)

Many candidates faced difficulty in tracing the recursive algorithm.

Binary Tree and Linked List (Question 17)

Many candidates lost marks by not reading question thoroughly.

The levels of knowledge, understanding and skill demonstrated

There is a wide range in the knowledge base of candidates who sat this examination. Many candidates wrote too briefly and did not develop their answers fully or accurately enough to earn all marks available.

The syllabus coverage seems to be good by most schools.

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

SECTION A

Question 1: Requirements Specification

Surprisingly some candidates had difficulty stating two items that would be included in a requirements specification, but in general the question was answered well.

Question 2: Hexadecimal and Binary Number System. Overflow Error.

Most candidates answered this question well. Some candidates did not understand the meaning of the action verb ‘state’. They wrote explanations on the error without stating the error condition.

Question 3: Functions of Arithmetic and Logic Unit

Generally this question was well answered. Most candidates knew that the ALU performs arithmetic and logical operations. Some candidates mentioned that it receives data to be processed, uses logic gates to perform comparisons and calculations, and returns the answer to accumulator.

Question 4: Magnetic Tape and Flash Memory

Most candidates answered this question well as tape is less expensive per gigabyte than flash media, has higher capacity and sequential access.

Question 5: Private and Public Class Members

This was a well answered question. Most candidates described that private class members can only be accessed from within the class in which they are defined and public class members can be accessed from outside the class.

Question 6: Use of a Macro within an Application

This was either answered well or scored zero depending on whether the candidates had come across macros as part of their exposure to IT.

Question 7: Checksum and Date Integrity

Many candidates confused checksum with parity.

Network protocols use checksums to detect errors in data transmission such as toggled, duplicated or missing bits. The sender calculates a checksum of the data and transmits the data together with the checksum. The receiver calculates the checksum of the received data using the same algorithm. If the received and calculated checksum do not match a transmission error has occurred.

Question 8: Prototype

A very well answered question – evidently result of the classroom theory backed up with the practical work on Program Dossiers.

Question 9: Interrupts

A few candidates wrote good answers with a clear explanation of an interrupt. Most candidates earned average marks as they understood the general idea but not the detail.

Question 10: Use of Keywords in Internet Search Engines

Most answers were general and subjective based simply from casual exposure to search engines.

Question 11: Serial and Parallel Transmission of Data

Some candidates did not explain their answers completely. Some candidates confused serial and parallel with simplex and duplex.

Question 12: Virtual Memory

Most candidates understood the general idea that virtual memory is a way of appearing to extend primary memory by using part of secondary memory in same way as primary memory but often they did not include enough detail to earn full marks.

SECTION B

Question 13

This was a well answered question although most marks could be obtained from general knowledge.

Most candidates outlined one method of data collection and suggested how this data collection would influence the design of the bridge. Use of sensors to count the amount of the traffic and the use of questioners at different points on the route to find out how many vehicles would use the bridge were answers that frequently appear.

Question 14

This question was answered well with sound knowledge of the topics being tested.

Most answers on sequential and binary search were correct.

Most answers on advantages and disadvantages of using a direct changeover were correct and complete.

Question 15

Most candidates correctly generated a long Boolean expression but were unable to provide a simplified Boolean expression. Some candidates did not attempt to construct the logic circuit.

Question 16

Tracing of the recursive method proved difficult to a number of candidates.

When attempting to identify the advantages and disadvantages of recursion candidates tended to write generally rather than give specific details.

Question 17

Many answers were excellent but some candidates did not read this question thoroughly and missed '...showing the information which need to be held...' and therefore omitted information to be held in each node of the tree.

In Part (c) many students wrote answers that were too vague and off course.

Question 18

Many candidates lacked depth in understanding. Many answers contain general points and less computer science.

The type of assistance and guidance the teachers should provide for future candidates

Many candidates could have earned more marks by carefully reading the questions to ensure they are aware of the command term used and noting the marks available to ensure the depth of the response is appropriate. The more examination practice the candidates get the better as it makes them familiar with how the questions are structured and the typical vocabulary used in the examination.

Higher level paper two

Component grade boundaries

Grade:	1	2	3	4	5	6	7
Mark range:	0 – 16	17 – 32	33 – 38	39 – 48	49 – 57	58 – 67	68 – 100

The areas of the program that proved difficult for candidates

The file-handling question (Q3) proved the most difficult of the four questions with the average mark being well down on the other three. As the question covered various methods of file-handling, this suggested a general weakness in this topic.

The manipulation of linked lists (Q2) proved difficult for some, but there seemed to be more problems in some schools than others.

The Case Study question proved more difficult this year with the introduction of the new Case Study that placed a greater emphasis on the concepts addressed in the HL extension, but scores were still reasonable on the whole.

The levels of knowledge, understanding and skill demonstrated

The candidates' responses did not always reflect the level of questions asked. For example, they did not always pay attention to the number of marks awarded or did reply accordingly to "discuss" questions.

The students' ability in responding to algorithm questions was generally good, and has shown a steady improvement throughout the last few years.

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

Question 1

This was well handled by the majority of students and extremely well answered by many. The students not only seem comfortable with the basic structures such as branching and looping, but also with the manipulation of objects, as shown by the use of dot notation in parts (b) – (d).

(a) Most students could describe the function of a constructor.
(b) Again. Most students were familiar with the use of dot notation.
(c) The method compare gained full marks for many students – some failed to break out of the loop if the position was found.
(d) There were many good solutions here including ones written with one loop only. Some students incorrectly copied over the initial array and then simply wrote in the new value in its correct place.
(e) Many students shuffled the elements but some put the steps in the wrong order and overwrote elements.

Question 2

Almost all candidates were familiar enough with linked lists to be able to answer on the descriptive level. As always, the coding of dynamic structures differentiated between the students.

(a) Most students picked up some marks here by referring either to the structure being dynamic and therefore easily expandable, and/or the ability to manipulate pointers.

(b) Again, this was well-answered. To gain credit for the diagram(s), the students had to clearly show how the pointers had changed – just showing a final structure was not enough.

(c) For students who had spent time studying linked lists this was a straightforward traversal which gained full marks for many. Some even produced recursive solutions. There are still schools which are weak when dealing with the teaching of dynamic structures. Some students did not attempt this. Some failed to use a temporary node to traverse the list and would consequently have lost access to the head of the list.

(d) This is where the more able programmers excelled and there were many of them. Common mistakes in those that attempted the question were the failure to deal separately with head of the first list and the failure to make the node added to the end of list B point to null. There were a variety of correct solutions.

Question 3

Questions on files continue to prove to be the difficult for many students. This was by far the lowest scoring question on the paper. As the question dealt with several aspects of files, it would seem that many schools are either not dedicating enough time on this topic, or that teachers should look at different ways of approaching it.

(a) How to create a partial index would seem quite straightforward but many answers were vague or confused.

(b) Several students provided static structures instead of dynamic ones and not many were are to clearly describe how it would be used in (c). The whole concept of why data structures are actually used in the memory during the running of a program is unclear with many students.

(c) See above.

(d) Again. Some answers showed a lack of basic understanding, but most gave the idea of needing extra storage space as a disadvantage of a full index with (overall) faster access being an advantage.

(e) Many had the idea of direct access but not all provided a clear explanation.

Question 4

The Case Study question didn’t prove as easy as in previous years. Schools should note that the more successful students are those who have prepared well for this question by studying in detail the Case Study in advance. It was clear when marking the question who had studied beforehand.

Questions that caused problems were:

(c) Some dealt with on-line systems rather than real-time ones.

(e) Some did not discuss **both** sides of the argument regarding access to information.

(f) Not many came up with good answers for the need for multiple fire-walls. The most common answer was a vague “more secure”.

(h) Many students confused “data” with “data fields”, and incorrectly said that data was inherited from the common Flight object. Not many mentioned that methods could also be inherited.

(i) Clients and servers were confused for actual people.

The type of assistance and guidance the teachers should provide for future candidates

Sufficient time should be dedicated to the Case Study. It was clear when marking which students had studied it in detail beforehand.

Be careful that the improvement in dealing with algorithms doesn’t come at the expense of the general theory.

Think carefully about how file-handling is taught. This is a difficult topic and one that can’t be treated fully within the programming course. Students should at least be clear about the difference and reasons for holding and using data in the memory and storing data in secondary storage.

Make sure that dynamic data structures can be manipulated without the help of outside libraries.

Standard level paper one

Component grade boundaries

Grade:	1	2	3	4	5	6	7
Mark range:	0 – 11	12 – 22	23 – 30	31 – 37	38 – 43	44 – 50	51 – 70

The areas of the program that proved difficult for candidates

Many candidates seemed particularly weak in their understanding of computers in dedicated, hardware control applications as opposed to the general-purpose computers such as a PC. Additionally, the implementation aspects of networking, selection of media, distance considerations, error detection and control etc., seemed to confuse a great many students.

The levels of knowledge, understanding and skill demonstrated

Outside of the areas identified above, the levels of knowledge and skill demonstrated on the examinations were appropriate for students at this level.

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

SECTION A

Question 1: Requirements Specification

This question was generally well-answered

Question 2: **File Format**

a. A few students choose inappropriate formats, such as bitmaps.
b. Many students simply stated something like “It’s compressed” without outlining how that is advantageous.

Question 3: **Binary Number System**

This question was generally well-answered

Question 4: Functions of Arithmetic and Logic Unit

Many students simply stated the functions named (arithmetic and logic) without outlining any element of the ALU’s operation.

Question 5: File Characteristics

A surprising number of students described searching for files in a directory rather than records within a file.

Question 6: Microprocessors

This was a generally well-answered question, although some students chose applications that were too general, such as “a PC”.

Question 7: Private and Public Class Members

This was a well answered question.

Question 8: Use of a Macro within an Application

Many students appeared to have never encountered or used a macro. Those who had, generally answered the question well.

Question 9: Check Sum

Many students simply stated what a checksum could be used for rather than describing how it would be used.

Question 10: Magnetic Tape and Flash Memory

Many students failed to make their comparisons in the context of backing-up data. Probably a poor question: Most students have no experience with or exposure to magnetic tape.

SECTION B

Question 11

Generally well-answered, but many responses were very vague and rambling:

a. Most students gave correct answers.
b. Many students erroneously believed that the item's information was in the barcode instead of the barcode being simply a key used to access the item's information.
c. Generally well-answered.
d. Most students were able to give plausible reasons.

Question 12

Many students struggled with this question:

a. This question was frequently misinterpreted by students wanting to distinguish between LAN's and WAN's. Comparatively few identified appropriate transmission media.
b. Many of the security measures suggested in answers did not provide the protections and access described in the problem as being needed. Few students achieved full marks.
c. Many students devoted a great deal of space to describing how a student would do something via the server but never explained what the server's role was in allocating and validating the use of particular resources.

Question 13

Students seemed to get this question largely correct, or mostly wrong:

a. Most students were able to complete the trace table.
b. Most students identified one of two errors but many of them described it as a runtime error, which is what occurs, instead of identifying the type of error (logical). Marks were awarded for either answer.
c. Generally well-answered.
d. Generally well-answered.

Question 14

This question was generally well-answered except for the identification of the advantages of producing more than one prototype.

The type of assistance and guidance the teachers should provide for future candidates

Future candidates would be well-served by placing additional emphasis on computers, especially microprocessors, in embedded applications in which dedicated hardware sensors, indicators, and actuators provide the inputs and outputs of the system and contrasting these with general-purpose computers.

Many candidates could have earned more marks by carefully reading the questions to ensure they are aware of the command term used and noting the marks available to ensure the depth of the response is appropriate. The more examination practice the candidates get the better as it makes them familiar with how the questions are structured and the typical vocabulary used in the examination.

Standard level paper two

Component grade boundaries

Grade:	1	2	3	4	5	6	7
Mark range:	0 – 10	11 – 20	21 – 30	31 – 36	37 – 43	44 – 49	50 – 70

The areas of the programme that caused difficulties for candidates

Whilst the majority of students showed an improved facility in the construction of algorithms it is still the case that some students are unable to tackle this aspect of the course with confidence. Students are expected at SL to manipulation array data structures and to write algorithms to solve complex problems.

In Question 2 whilst many students showed an understanding of the relationship between a master file and a transaction file they were not able to clearly explain why both needed to be in the same order (Q2(c)).

In general, many students did not use the appropriate symbols to construct the systems flowchart, however, where able to show their top-level understanding of the update process.

In respect to the Case Study, whilst in general well answered, understanding of Real Time

Processing as compared to Online/Interactive Processing was somewhat lacking.

The levels of knowledge, understanding and skills demonstrated

Candidates showed a pleasing ability to construct algorithms and to handle manipulation of data stored in an array, define the return data type and were also able to handle an array of type record as defined by the Player class in question 2.

The relationship between a mastery file and a transaction file was well understood.

The case study was in general well answered.

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

Question 1

a) Most students understood that an array was required and that the type was Boolean, however, many students did not comment on the convenience of the processing of such a structure.

b) Generally well attempted. Students initialised the counter and also showed a good understanding of the need to define the return data type and to include the return statement.

c) Reasonably well answered. It is important that students appreciate that data structures and variables need to be declared and initialised. The loop structure was well answered and many students were able to handle the nested if statement and include an appropriate return statement. This is a complex algorithm and it is pleasing to see that students are able to tackle such problems. Problems of this standard will continue to used to assess students ability to manipulate array structures.

d) Students need to be aware that methods can only return one data value or a reference to an object. In this case many students suggested the use of an array with two values.

Question 2

a) Most students answered this question well and showed a clear understanding of the difference between a master file and transaction file.

b) Many students showed the logical flow of the update process but did not use appropriate symbols for the System Flowchart.

c) In general this question was poorly answered. Students at SL are expected to understand issues of efficiency in processing. In this case the efficiency is improved if the same order is used. A number of students incorrectly suggested that the updating would be inaccurate if the order was not the same.

d) This part of the question was well answered.

e) A difficult algorithm that was in general well answered by those students who attempted it. A key aspect of the algorithm was the efficiency of the inner loop. It is expected that students at this level consider such issues when constructing solutions to be awarded high marks.

Question 3

In general the Case Study questions were well answered

a) i) Candidates showed a good understanding of parallel running.

ii) This part of the question was well answered.

b) Many students showed that they understood the concept of testing in a realistic manner which was pleasing.

c) The examiners were looking for two different methods. Many students provided two biometric examples.

d) i) Many students did not understand the concept of Real-Time processing and confused it with online processing.

ii) For students who correctly identified the real-time processing this question was well answered.

e) i) This part of the question was well answered.

ii) The concept of User-Interface was reasonably well understood and many students suggested some form of touch screen as the most appropriate.

iii) This part of the question was well answered

iv) Reasonably well answered; candidates clearly understood the importance of a back-up strategy.

f) Most candidates considered the important role the Government plays in ensuring safety and the conflicting demands that this may make on customer privacy. The question was worth 6 marks and many candidates failed to discuss the different points in sufficient depth. Often candidates either simply stated a point without any justification or commented in a general way without specifics.

The type of assistance and guidance the teachers should provide for future candidates

Students need to be encouraged to read questions carefully, practice writing algorithms and prepare for the Case Study.

Teachers need to ensure that students are able to write System Flow Charts using appropriate symbols.

Students need to be provided with more opportunity to write extended responses such as is expected in Q3 (f)