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 - 72	73 - 100

Standard level

Grade:	1	2	3	4	5	6	7
Mark range:	0 - 15	16 - 31	32 - 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

Nearly all HL Dossiers submitted addressed an appropriate problem to enable the HL mastery factors to be meet, however, in a few cases it was noted that candidates addressed mostly the SL mastery factors. Schools and teachers need to check that the HL mastery factors are clearly understood and that program code addresses these at the appropriate level.

It was pleasing to see that nearly all problems solved were realistic with a real client. Some students are submitting game problems; caution should be taken to ensure that a real client exists and that the mastery factors can be adequately addressed.

Candidate performance against each criterion

Criterion A1

Well done; students showed good data collection and analysed the problem well. It is very important that students show that they have collected realistic data about the problem: interviews, observations, existing documentation etc.

Criterion A2

This section is crucial. Most students did outline reasonable objectives, but some tended to be vague. The students need to keep in mind two factors: relate the objectives to the analysis and also ensure that the achievement of the objectives can be assessed in the testing.

A simple table with four columns may help: objective number, statement of objective, rationale (link to A1), how achievement determined.

Such a structured approach forces the student to address the latter two factors.

Additionally, A2 is linked to C3 and the table can be reused with the addition of a statement for each stating how it was achieved - this can be the test table and also linked to the hard copy section (D1) by way of hard copy reference number. A detailed A2 section also makes it easy to address the report in D2.

Criterion A3

Most students produced good prototypes, included convincing user feedback and suggested changes that needed to be made to the design.

One aspect that students need to focus on is presenting some form of initial design flowing from the analysis.

This does need to be in the form of the modular design in B3, but it should show some level of detail related to inputs and outputs.

Criterion B1

This section was reasonably well done. Many students considered memory based structures along with disk based structures, justified these and showed how operations would be applied using sample data.

It is important that the student justifies the data structures used, and it is a good idea to also briefly address relevant mastery factors.

Classes/Objects and Class/Object structures need to be included here.

Criterion B2

This section seemed to have improved with fewer students showing obvious code as algorithms.

Teachers are encouraged to keep at it and get students to construct algorithms. These can be modified as is needed during the development process, but they should not be completed after the event.

Criterion B3

This section also seemed to be much improved. Simple modular designs showing connections with some explanation is all that is required.

There is no particular standard required. Students are free to adopt which ever approach they feel confident with.

It is important to show connections e.g. class diagrams or structure charts, and include some detail about how the linking operates.

Criterion C1

Code listings are often full of IDE generated GUI code. The students should include indications in the code as to what is theirs and what has been generated. It is also a good idea to mark clearly where and state why a mastery factor is claimed.

Students are expected to use header documentation as well as inline comments, use meaningful names and set the program out in a consistent manner to show its structure.

Criterion C2

This was reasonably well done. Students should consider all user I/O potential errors, file I/O and other types of error that can be trapped.

A table is a simple approach, with some kind of statement about why the testing is sufficient.

Criterion C3

This section really should be formally included as a simple table by reusing the A2 objectives table and stating why each objective has been achieved.

A testing table with hard copy reference numbers is a useful way to link to D1 and to also show that the hard copy has been comprehensive.

DOCUMENTATION

Criterion D1

There seemed to be an improvement in the comprehensive nature of the outputs provided. It is very important that the student shows that the program functions as intended and does this in a convincing manner.

Criterion D2

This section also showed improvement with students addressing all the areas.

Recommendations for the teaching of future candidates

Schools and teachers need to:

- Ensure the new criteria are used
- Ensure that the documents on the OCC are accessed and read

Mastery Factors

The overall issue is to make sure that the claim is based on evidence and is justified e.g. just using recursion when iteration would suffice and be more efficient does not necessarily mean the mastery claim will be awarded.

Many schools and teachers did not claim the mastery factors related to the 5 SL mastery factors and use of additional libraries. At HL it is nearly always appropriate to claim these.

The following mastery factors are still often not applied correctly:

- When using a RAF, the student needs to show manipulation of the pointer e.g. seek(recordPosition * record Length) rather than simply appending. One way to do this is to randomly assign the position to write to, and check if it is flagged for deletion or is occupied before writing the new record or over-writing the existing record to update it.

- Recursion is often trivial or incorrectly implemented.
- Polymorphism using overloading and over-riding is allowed, but there ought to be a justifiable reason. Likewise, multiple constructors is appropriate but often not for a default and one other.
- Encapsulation needs to use private instance variables.
- Parsing a text file is still misunderstood with a number of candidates claiming the use of parseInt. The intention is that students read in a data stream and split the steam up in segments which are assigned - StringTokenizer does this.
- ADTs continue to present problems. Students need to justify their claims. It is also not possible to use Java ADT e.g. the LinkedList class to claim mastery.
- The Hierarchical Composite Data Structure claim is one that presents problems, and teachers and students need to review what is specifically meant by reading the description carefully.

As stated in last year's report, it is important that students do not plagiarize code without acknowledgment and also do not use this code to claim mastery.

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

There has been a steady but noticeable improvement in the general standard both of dossier presentation and execution. The selection of problems and the presentation of dossiers in this session were generally good or very good. There are always some unsuitable choices made by candidates. Typically these were unable to achieve the required mastery aspects or there was no real user and this prevented the candidate gaining a solid understanding of the nature of the problem. Needless to say, if you don't understand a problem, it is that much harder to solve it successfully.

A handful of dossiers were large in scope and tried to achieve too much in the time allocated. At SL a quality document of about 50 pages will be adequate. The information problem should be real but simple (or simplified).

Students who chose an end user who was close and accessible (such as a teacher or close relative) found it much easier to gather the required data for the analysis and goal-setting parts.

Mastery factors should be considered at the beginning of the process and teachers/students should have in mind at least 10, preferably 12, that they can see will most likely be achieved within the scope of the problem. A serious effort at prototyping will help this aim.

Teachers must be sure they are using the correct version of the Subject Guide and should follow the structure indicated there.

Candidate performance against each criterion

ANALYSIS

As always, the candidates who performed poorly in this section or who wrote it last also performed poorly in the rest of the dossier. This is the solid foundation on which a good dossier is constructed.

Criterion A1

Students often assume knowledge of the problem domain on the part of the reader and should be encouraged to give a general introductory background.

There must be evidence of data-collection. In an ideal case, a small manual system is being improved and existing documents from this system can be invaluable in providing sample data. Photographs, interview transcripts and questionnaire results are all good examples.

Criterion A2

Students must relate the goals to criterion A1 explicitly. This implies some sort of explanation as to why each goal is important. For example:

> "*There must be a way to edit a user's profile in the system. From the interview results it is clear that this feature is important to the end users of the dating system."*

About 6 goals or objectives of this type can create a project of a good scope. 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.

Generalizations referring to "user friendliness" should be avoided as this would normally be expected anyway and are too hard to evaluate and measure success.

Criterion A3

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 showing the user what they can expect to see. End user feedback should be presented.

DESIGN

The design section can cause problems and this is often due to a poor analysis of the problem.

Criterion B1

Time spent on design is time saved in the long run as the history of system development clearly shows. Encourage students to be thorough in this section and to use clear diagrams and tables in place of wordy and vague explanations. This will also help sharpen their thinking about the problem and potential solutions.

Criterion B2

Very detailed algorithms are not really necessary here and can be a lot of work for the marks allocated. Candidates must provide parameters and return values plus a good description of the major, important algorithms. One way to describe an algorithm is to use some form of modified code, pseudo-code or semi-formalised English.

Criterion B3

Almost all schools are now encouraging an OOP approach since this is now well into the mainstream of code design. Therefore this section should probably be dealing with classes, showing links to data structures via the instance variables of the class and links to algorithms via the methods (setters and getters would be trivial and need not be detailed anywhere).

THE PROGRAM

Criterion C1

Moderators must read many lines of code each session; therefore the wise student will make this process easier for them. Numbering pages and lines, using a mono-spaced font, starting a new class on a new page, using landscape orientation to avoid line wrap and liberal comments to separate methods are all helpful.

As a teacher you can help students by setting conventions, such as those above, and refusing to spend time looking at code which does not conform.

Criterion C2

This has improved in recent sessions with many candidates providing code examples by either quoting code or referring to specific methods and lines precisely.

Criterion C3

It is not necessary to produce additional documentation for this section as D1 should demonstrate how well the solution actually worked and D2 should provide an evaluation. However, the student could take a checkbox approach here just to satisfy themselves as to which A2 criteria have been met.

DOCUMENTATION

This section should, at least initially, focus on the candidate’s achievements.

Criterion D1

Candidates should carefully pick screenshots that show that all claimed mastery aspects are working and which demonstrate that each criterion in A2 has been achieved. More sample runs should be made with valid data than with invalid data. This continues to be a problem for most candidates who prefer to fill the dossier with pages of error messages or invalid runs.

Evidence must be presented for the award of mastery aspects.

Criterion D2

This section too has shown a great improvement in quality in recent sessions and it is good to see candidates involved in critical reflection of their achievements. This task has, in almost all cases, been very challenging so it is good that they get value out of it.

Recommendations for the teaching of future candidates

A common theme is the lack of evidence of a working solution, coupled with poor code listings and a lack of organisation. The poorest dossiers are all but impossible to mark highly yet some teachers continue to do this.

Please encourage even the least able of candidates to document what they have achieved via screenshots which are annotated. Moderators cannot confirm high marks where this has not been done.

Where simple "setter" and "getter" methods and constructors are used as part of an Object the same methods should not also be used to claim mastery of methods with parameters; methods with return values as such methods are usually trivial.

Some SL candidates are not being awarded mastery of flags/sentinels or use of additional libraries when it appears they could have been. Additional libraries, for example GUI libraries, are utilities such as StringTokenizer, ArrayList or LinkedList. Mastery of additional libraries should not, however, be claimed for using java.io for File Handling.

Higher level paper one

Component grade boundaries

Grade:	1	2	3	4	5	6	7
Mark range:	0 - 15	16 - 30	31 - 39	40 - 49	50 - 59	60 - 69	70 - 100

The areas of the programme that proved difficult for candidates

- Questions 1, 2a, 7, 8, 12, 15, 18d and 18e proved difficult to candidates.
- Question 1 generated answers with only one way in which the functioning of a compiler differ from an interpreter.
- Question 2 was hit and miss as to whether or not students had come across OCR software during the course of their general IT exposure.
- Question 7 - Most candidates were able to explain only one way of reducing time required to transmit data in a computer network. Most students answered that high speed lines could be used. Only some mentioned data compression.

- Question 8 - Below average marks earned. Many students were not able to define the term truncation error.
- Question 12 - Handshaking and polling - it seems that many candidates have not learnt these terms.
- Question 15 - Part (a) was not well answered. Below average points earned.
- Question 18d and 18e - Many students did not attempt to answer these questions.

The areas of the programme in which candidates appeared well prepared

The performance of most students was reasonable and acceptable.

There were many excellent students and a few students who were very poor in their performance. Many candidates write either brief or too long answers and tend to neglect the amount of marks assigned to each question.

The syllabus seems to be covered well by most schools.

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

SECTION A

Questions 1, 2a, 7, 8, 12, 15, 18d and 18e were poorly answered. Please see above.

Questions 4, 6, 9, 13 were answered better than expected.

SECTION B

Question 14 - System Analysis

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

Question 15 - The roles of registers and buses in operation of instructions

Part A was poorly answered. Part B was well answered. Many students were able to describe correctly the role of buses in fetch-execute cycle.

Question 16 - Recursive binary search

This was the best answered question.

Question 17 - Hospital computer system - types of processing

This was a generally well answered question. Some students confused real-time with on-line processing.

Question 18 - Fixed and floating point number representation

Parts a, b and c were well answered. Many students did not attempt to answer Part d and Part e. Some students who attempted these questions did not show the work out to earn at least partial marks.

Question 19 - Packet switching. Standard protocols

Most candidates understood the use of standard protocols and were able to outline how packet switching provides better security for data being sent and to describe how packets are correctly re-assembled by the receiving computer.

Recommendations for the teaching of future candidates

Exam practice - many students could have earned more marks by being less subjective and general. Questions should be analyzed in order to determine what is being asked. The number of marks available is equal to the number of points to be made. All points 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 - 16	17 - 32	33 - 38	39 - 48	49 - 57	58 - 67	68 - 100

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

Traversing dynamic structures is normally required in most sessions, but this continues to prove difficult.

Hashing is still a technique that is not completely understood.

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

The trend in increased understanding of objects and their use that has been shown over the length of this specification continued. It is clear that teachers themselves have improved their own teaching of this topic, with many, perhaps, teaching OOP from the very start of the course.

The students also appeared well-instructed in showing diagrammatically how nodes are inserted.

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

Question 1

- Most students had the understanding that private variables protected in some way the values stored within them
- Use of the constructor was well-understood

- There was a wide-spread use of getter / setter methods
- 2D arrays were generally handled well

Question 2

- Most showed clear linked-list diagrams
- Adding a node to the linked list as always produced a wide range of marks, differentiating between those who understood the theory and those who didn't
- BigO values were generally correct, with many also showing good understanding by picking up the explanation mark.

Question 3

- Sequential files gave an easy lead in to this question, but students were less sure of how both hashing and index files worked. The idea that a single file could have more than 1 index did not seem to be well-understood
- Advantages over dynamic over static and trees over lists were handled well

Question 4

This Case Study now requires more research to be carried out than previously. This trend will continue with the new Case Study ("Smart Phones"). It is becoming increasingly more difficult to gain marks on this question just by simple reference to the study during the examination.

Recommendations for the teaching of future candidates

- Practice the standard ways of traversing dynamic structures, although this is not a topic that can learnt without a reasonable understanding of the theory
- Look carefully at the areas in the new Case Study which require additional research on the part of the student

Standard level paper one

Component grade boundaries

Grade:	1	2	3	4	5	6	7
Mark range:	0 - 10	11 - 20	21 - 25	26 - 31	32 - 37	38 - 43	44 - 60

General comments

The candidates writing the May 2011 Computer Science examinations displayed a good understanding of the subject overall. Notably, the number of papers demonstrating very poor understanding seemed to markedly decrease from previous examinations. The increasing depth of understanding displayed with regard to object-oriented concepts and social issues related to Computer Science suggests increasing abilities amongst teachers to effectively prepare their students.

Areas of the programme and examination in which the students appeared well prepared

The trend in increased understanding of objects and their use that has been shown over the length of this specification continued. It is clear that teachers themselves have improved their own teaching of this topic, with many, perhaps, teaching OOP from the very start of the course.

Candidates appear to be developing a strong grasp of the social implications and limitations of Computer Science.

The levels of knowledge, understanding and skill demonstrated

The performance of most students was reasonable and acceptable.

There were many excellent students and a few students who were very poor in their performance. Many candidates write either too brief or overly long answers and tend to neglect the amount of marks assigned to each question.

The syllabus seems to be covered well by most schools.

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

Question 2

Many candidates failed to follow the explicit direction that the answers were to be 6-bit numbers.

Question 10

Many students described the process in (b) in such vague, non-specific ways that it was not possible to discern that any part of the process was related to the server and thus received rather few marks.

Question 11

This was generally quite well-answered. In particular, candidates demonstrated a great deal of sensitivity and insight in their answers to part (c).

Question 12

This contained an error that resulted in significant confusion. The question was removed from the examination and candidate scores were based on the reduced number of marks available as a result.

Recommendations for the teaching of future candidates

Exam practice - many students could have earned more marks by being less subjective and general. Questions should be analyzed in order to determine what is being asked. The number of marks available is equal to the number of points to be made. All points should contain computer science and not general observations.

Standard level paper two

Component grade boundaries

Grade:	1	2	3	4	5	6	7
Mark range:	0 - 12	13 - 24	25 - 32	33 - 38	39 - 44	45 - 50	51 - 70

The areas of the programme that caused difficulties for candidates

Students on the whole showed an improved understanding of the construction of algorithms that dealt with arrays. However, there are still a number of students who either are unfamiliar with arrays or not confident with the need to use an index to access data elements.

Students need to pay careful attention to the marks allocated to a question and ensure that they answer accordingly. As an example, question 2(a) was awarded 2 marks and students were expected to outline two distinct differences. Many answered whole numbers vs decimal numbers and did not add an additional point relating to the memory storage differences.

Many students were not able to adequately explain the notation of creating a class and why constructors need to be public. Whilst OO concepts as described in the HL syllabus are not examined at SL it is expected that students understand the process of instantiating (creating) a class and the role of the constructor.

SL students should also know how to use dot notation and be able to handle arrays of type record object as required in Q2. Many students demonstrated difficulty in these areas.

The bubble sort question was not well answered with many students not describing a bubble sort but some other sort, and a number failed to outline 6 distinct points to be awarded full marks.

The case study was on the whole well answered but it is important that students be specific in their answers and relate points made to the case study itself.

The levels of knowledge, understanding and skills demonstrated

The general level of knowledge of algorithms and arrays was pleasing.

The case study was well answered and students showed good ability to answer written questions.

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

Question 1

This was well answered. Students showed understanding of determining the operation of an algorithm and also good ability to construct logical algorithms to solve problems related to arrays.

Question 2

Students handled the algorithm aspects well. As mentioned above there is a need for students to spend some time practicing dot notation and handling arrays of type record object. Many students showed a very good understanding of the idea of a method return value and how to use this to index an array.

Question 3

The case study questions on the whole were well done.

Recommendations for the teaching of future candidates

- Students need to be familiar with array processing and should be given plenty of practice. This should include indexing arrays for different purposes, understanding how to return values from a method and how to use arrays of type record or object as outlined in Q2.
- Students need to be made more aware of the need to match the number of points made in a response to the number of marks allocated to the question as a whole.
- Students need to be encouraged to read questions carefully, practice writing algorithms and prepare for the Case Study.