November 2017 subject report

Computer Science

Overall grade boundaries

Higher level

Grade:	1	2	3	4	5	6	7
Mark range:	0 – 16	17 – 34	35 – 45	46 – 54	55 – 63	64 – 72	73 - 100

Standard level

Grade:	1	2	3	4	5	6	7
Mark range:	0 – 19	20 – 38	39 – 49	50 – 57	58 – 66	67 – 74	75 - 100

Internal assessment

Component grade boundaries

Grade:	1	2	3	4	5	6	7
Mark range:	0 – 5	6 – 10	11 – 15	16 – 19	20 – 23	24 – 27	28 - 34

The range and suitability of the work submitted

The vast majority of candidates described diverse scenarios that allowed for worthwhile projects. As opposed to the May session, the majority of solutions were based on database projects using either MS Access or MySQL/PHP. Conversely, a few schools submitted a variety of products. It is hoped that the range of solutions continues to expand in future sessions.

The quality of the solutions showed a wide range and not all solutions had been developed to the level of complexity expected of IB DP candidates. Some examples of trivial products included; Java programs that mainly focus on the GUI rather than the functionality, Java programs that consist of one class only, rudimentary versions of freely available games (like Sudoku), Access databases that contain less than three tables or tables that are not joined, Access databases that only include rudimentary queries and macros, websites that are template-based (Wordpress, Wix or Weebly) or that have minimal content, basic Excel projects and Scratch projects that have not been properly designed.

Highly successful solutions tended to incorporate features from more than one type of software. For example, website projects that incorporate JavaScript / PHP / SQL functionality, or programming projects that interact with sensors, Access databases or with on-line resources.

The required video component has become more structured as many candidates address the Criteria for Success to show how functionality has been achieved.

Candidate performance against each criterion

Even though most products were of the intended level of complexity, students typically made superficial attempts at documentation and explanation, and consequently lost marks.

Criterion A: Planning

This criterion has seen a steady improvement over the years. Some candidates, however, still do not follow the expected sequence:

- investigate a situation
- identify client/adviser
- explicitly consult the client (and/or adviser)
- describe the scenario with explicit reference to evidence of the consultation added in an appendix
- choose a solution

- describe the rationale for the solution and also for the software to be used
- identify appropriate Criteria for Success for the chosen solution.

Too many candidates decided on a product ('I want to make a website/program a game') and then found a client to match. Contrived tasks and clients were routinely seen in the weaker candidates' work. Most reports show some evidence of client interaction in an appendix, but more often than not, it is very limited in content and it is not being referred to in the description of the scenario. Many candidates had generic success criteria – these criteria must be specific and testable. The Criteria for Success are essential to the project and must be explicitly addressed in the test plan (Criterion B) and in the evaluation (Criterion E) and preferably also in the screencast.

Criterion B: Solution overview

The quality of the solution overview has decreased a bit with many projects showing a superficial effort to design the solution. Specifically Database projects require some consideration of normalization and design of complex queries/macros in addition to table design and form design in order to reach the highest descriptor. Records of Tasks were generally only partially complete, typically because the final product had not been implemented by the client. A wide variety of test plans were seen. The better ones aligned with the Criteria for Success.

Please note: The use of the proper template in forms.zip is mandatory – the use of a different version, with incorrect columns and headings, will be penalized. If no Record of Tasks is included or if there is no evidence of a design then 0 marks will be awarded.

Criterion C: Development

Most candidates made a good attempt to document the development of their product and the techniques used. However, the quality of the explanations and the completeness of techniques typically left something to be desired. The complexity of the product must be justified by the candidate in the write-up. A seemingly complex product without proper explanations of complex techniques used in the product, only achieves moderate complexity. Similarly, high ingenuity must be justified by algorithmic thinking (e.g. explanations of complex data structures, algorithms or macros).

Criterion D: Functionality and extensibility of product

The screencast should only show the proper working of the solution as outlined by the Criteria for Success. Some screencasts focus instead on the development of the solution, which made them too lengthy. Others only show the working of the interface, without showing actual functionality of the intended solution. Extensibility was impacted by a limited Design Overview and the failure to include a design version of the product (java vs jar files or a database accessible in design view).

Criterion E: Evaluation

This criterion has the most potential for improvement. Because many projects addressed a contrived need, there was limited client involvement and products were typically not implemented/used as a solution by the client. Subsequently, there is little opportunity for a worthwhile product evaluation and some students only ticked off the Criteria for Success without further discussion. For full marks evidence of meaningful client feedback against the Criteria for Success must be included (in an appendix) and it must be discussed and referred to in the candidate's evaluation against the Criteria for Success. Recommendations should be realistic in relation to the actual product – for example 'adding network capability' is not a realistic improvement for a low-level product.

Recommendations for the teaching of future candidates

The aim of the IA in Computer Science is to create a working solution for a real client. The consultation (included as an appendix) should be the basis for the description of the scenario, leading to a range of specific and testable Criteria for Success of a chosen solution.

Criterion B should provide evidence of a rigorous design stage with an overview of all five stages of the project (including the actual intended use of the product by the client) in the Record of Tasks, detailed layout design sketches that include annotations for complex techniques, evidence of algorithmic thinking (in the form of flowcharts, UML diagrams, pseudo-code, entity relationship diagrams, structured database decomposition using normalisation, query and macro design), and a test plan that addresses all Criteria for Success. All high scoring projects included a thorough design stage.

Criterion C provides candidates with the opportunity to demonstrate their knowledge and understanding of the tools and techniques used in creating the product. The use of tools/techniques should be explained in relation to screenshots and/or code excerpts that show their use.

Criterion D does not require written documentation. The screencast should be limited to 5-7 minutes and should only show the proper working of the final solution. For example, the screencast could show the testing of the implemented solution following the test plan from Criterion B. Successful screencasts showed comprehensive evidence of the solution's functionality with lots of data, but were edited to avoid viewing tedious data entry. Candidates are advised to test their screencasts on different media players and devices to ensure the playback is correct.

Extensibility is evidenced by a detailed design in Criterion B, by a detailed description of the creation process in Criterion C, by a product that is accessible in design format and, in case of a programming project, by a properly structured and annotated code listing in an appendix.

Criterion E should provide evidence of a rigorous evaluation stage. The client feedback (added in an appendix) should be discussed by candidates as part of their own evaluation of the solution. A table showing the Criteria for Success with a 'met/not met' evaluation is not sufficient to achieve in the highest descriptor. Recommendations for improvement should go beyond the success criteria that have not been met.

A word of caution: treating the project as a purely academic exercise typically means that there is no proper client and that the solution is not being implemented, which could potentially have an impact on Criterion A, Criterion D and Criterion E.

The recommended word count **for each section**, as indicated in the TSM, is only for guidance. The **overall** word count of 2000 words however, is a fixed limit and a moderator is not required to read beyond this limit. Failure to comply with this requirement could lead to the candidate self-penalising in Criterion E.

Further comments

For additional information regarding the Computer Science IA, please consult:

- Computer Science Guide (pages 56-72).
- Teacher Support Material (Internal Assessment) available on the OCC.
- Forms.zip templates.
- Submission of the Computer Science IA in the Handbook for Procedures for the Diploma Programme 2017 (Section B4.4). Note that the Handbook is updated yearly.
- IB Coordinator Notes.

For additional professional development regarding the Computer Science IA, please consider:

- Getting involved in the Computer Science OCC discussion forum.
- Registering for Computer Science workshops (either face-to-face or online).

Higher level paper one

Component grade boundaries

Grade:	1	2	3	4	5	6	7
Mark range:	0 – 19	20 – 39	40 – 52	53 – 60	61 – 69	70 – 77	78 - 100

General comments

Many of the comments in this paper also apply to SL Paper 1.

The range and scope of questions in Nov 2017 Paper 1 is comparable to those of previous sessions; however, for this paper several simplifications were made that delivered altogether a more accessible paper. The simplifications include a reduced amount of questions (and marks) on code construction, a structure that reduces the overall number of sub-questions, a better distribution of marks across Section A and B, with more opportunities to hit high scoring questions in Section B.

The large majority of candidates attempted all questions, confirming that the structure and design of the paper helped in time-management.

A number of responses scored highly, and these showed good abilities in programming, code analysis and transformation, basic factual knowledge, and application of that knowledge in simple specific contexts.

However, it was observed very often that many answers were simply developing off-course and could not even constitute a plausible answer to the given question: this was quite puzzling.

In most cases, a bit of attention and detail in writing, and the use of specialist terminology would have helped reducing the loss of marks, that was mostly generated as follows:

1) Schematic (tabular) style of writing is often insufficient to address the explanatory level expected in Assessment Objective 3 (AO3) questions.

2) Very short sentences (half a line) that are not enough to answer a question that is based on an AO3 command term (explain etc);

3) No use of specialist terminology, or improper use of operations proper of data structure understanding them just as basic English words rather that operations in computer science.

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

Most candidates attempted exercises on code construction, but many of them did not demonstrate a level that is expected for an HL candidate in Computer Science. Some attempts of responses were using a style of writing code that was insufficient to address any possible procedural reasoning; correct or incorrect.

The number of candidates that cannot write any reasonable form of code or any working loop is concerningly high. Some candidates that delivered poor responses in the simple code, decided to skip the more complex one and focus on other questions.

The ability of using some formal syntax (also simple mathematical operations) is concerning, with a many candidates using “>” instead of “<” or writing a loop in the style “while x >0, if x =0 do .. else if x /=0 do...”.

Different levels of abstractions in data structures are often mixed up, so that for some candidates an array is the same as a stack, which is the same as an object in Java (despite Java not being assessed in Paper 1) and therefore be subject to the Java methods of adding elements in collections (when collections are not asked for!). These facts, observed in a scientific subject, are evidence of little understanding and major confusion, rather than broad knowledge.

Very surprisingly some candidates cannot define “bit”, “protocol”, “client”, “server” either and elaborate answers showing little understanding of the values of those terms in Computer Science. Some rely on misconceptions, including, for example, the fact that in dynamical data structures, the size is fixed and cannot be altered at all.

While many candidates know how a stack works (not all of them though!), they can write imprecise answers that do not show much understanding of the operations of push/pop/is empty on the data structure itself, and just refer to those verbs in plain English. This is a major problem because as long as responses are written with that style, it makes very difficult to assess the ability of algorithmic and computational thinking. At least two responses manipulated a stack as if it were a queue and one manipulated a stack as if it were an array where one can cherry-pick in the middle.

Understanding the output of a simple recursive procedure has been challenging for many candidates as they only return the base case of the recursion.

Program transformation into an iterative version (as specified in the wording of the question) has been difficult. Many could test the parity of the input, but alone it was not sufficient to produce a working program.

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

Basic factual knowledge is generally OK. It appears that some candidates may have that knowledge but failed in expressing it with the level of accuracy (terminology and detail) expected at HL.

Many candidates know the difference between a “while” loop and a “for” loop (and which one is preferable in specific situations). However, sometimes they do not correctly set the loop boundaries.

Truth tabling construction is broadly well understood, with a few exceptions. Many responses used truth tabling also to answer the logical diagram (gates) question.

On logical diagrams, some responses were fully correct, and a large majority almost correct (losing marks for ignoring the priority of different gates).

Tracing is usually well received, with the exception of recursion, although with some imprecision in handling the output or with termination.

General factual knowledge on interrupt/polling, system implementations, societal impact and packet switching are usually fine, but they often lack the level of detail that address the elements of relevance for Computer Science.

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

1. Some candidates have lost marks by not answering the question, and addressing aspects on how useful computer languages are. Others have used non-specialist terminology resulting in lightly ambiguous wording, incurring into repetition of concepts.

2. (a)(b) Generally both fine, but marks lost for short answers without detail. Some incorrect answers indicating client and server in terms of a human customer /shop attendant or subscriber of internet services and ISP.

3. Generally OK. A few answers did not refer to ‘inputting devices” as required, though, rather to screen resolution.

4. Most candidates returned a table with 4 lines, usually correct. However, some candidates returned tables with 1,2, and even 8 lines.

5. A few candidates could do it right. Many returned a version (second example in the MS) that suffers for associativity. However, a number of strange drawings were returned, including those where AND or OR gates had just one wire in input.

6. (a) No problems.

(b) A few returned the full correct answer, many returned either 0 only, or the reverse complete sequence. Some returned only 2.

(c) Some candidates attempted it, but incorrectly, resulting in a NAQ that allowed to receive partial marks (typically a program that was outputting all numbers). Some candidates made it very wrong with a very bad use of loops or even preserving recursion. Only a few people made it right.

7. (a) Generally OK, but some answers were incorrect and other ambiguous, generating loss of marks.

(b) Generally OK, but some candidates referred to the 'instruction bus', or have used lengthy descriptions that did not provide evidence of understanding the role of the two specific buses (nor at which level of the system they are present). Responses that showed some general correct understanding without sufficient detail received partial marks.

8. Many candidates do not know what a bit is, in Computer Science, and are not able to define it. This includes answers such as: "a bit is 1/8 of a byte", "8 byte = 1bit", "a bit is much less than a Megabyte", "a bit is a tiny little portion of something".

9. Generally OK.

10. (a,b,c,d) Generally OK.

(e) Some candidates have attempted it using a very poor unofficial syntax for pseudocode, which led to major ambiguities at times. Others have attempted it with a better style, but might have used an output rather than a return. Many candidates decided to use an incremental for-loop which is not the best way to proceed for this specific question, and may have delivered a non-working solution that could have obtained only partial marks. Those who have used a while loop have usually done it correct, possibly losing marks with the loop boundaries or the end of array condition.

11. (a,b,c,d). Generally OK. However, some candidates were off course in part c and there were some vague responses in part d.

(e) Generally OK, but many responses were limited to the description already given in the figure in the question paper without further expansion. High number of responses that were just listing the many fields without saying what they were used for, how, at which stage of the process. A few responses were relating packed switching to communication within the CPU, linking therefore with the question on buses. Other responses were expanding with too much detail into the TCP/IP model, and ended up not answering the question.

12. (a) Many candidates went off course, repeating the merits of these scans for security (repetition with the stem in the question paper).

(b) Generally OK.

(c) Only a few candidates know what a transducer is, and many referred to an analog-digital convertor, that is “necessary for a digital image in input”. Some responses refer to sensors that were not even mentioned in the question paper. Some loss of marks of many. Several candidates know well what a transducer is and understand how it can be used in the given context.

(d) Generally OK for the majority. However, many candidates tried to establish a difference between a “centrally controlled system and a central system (sic)”.

(e) Some candidates cannot name two distinct operations of an operating system.

(f) The majority of candidates answered with the Interrupts and elaborated on that. A few chose polling with some elaboration. A large number of candidates simply described the two methods without taking a position and without even elaborating this in context: this style of answer is not enough to address the question for the way it was worded.

13. (a) Mostly answered correctly, but a few candidates wrote that the dynamic data structure has ‘fixed size”, others did not refer to node at all, and some others were lacking the necessary detail to get full mark.

(b) This question raised a huge variety in style in the answers; some of them returned a series of sketches (time consuming activity) without any written explanation on the process: they might have been correct, but they did not provide the needed evidence to address aspects of searching, and may therefore have lost a few marks. Apart a few answers that did not consider the relevance of alphabetical order, the question was in general well received.

(c) Several responses got the sequence incorrect, or even listing all the elements of the two stacks in some order, or even listing a varying number of Xs and Ys as output.

(d) Candidates showed a general understanding on how to proceed, but expressed this understanding with a very weak terminology, and with general terms, without making reference to the technical meaning of the operations ‘pop’ and ‘push’ on stacks, and therefore losing the needed (and sought!) algorithmic detail that we intend to observe. Some candidates swapped the order of the two stack, intensifying the general perception that these kind of questions are not attempted or answer with a correct ‘computer science’ mindset at all.

14. (a) Usually answered correctly.

(b) Mostly answered correctly.

(c) Some fully correct responses and several candidates skipped this entirely. Many attempted solutions that used a style of writing code that was either confused or too poor to be able to award marks to the elements we wanted to observe. Marks lost on incorrect loop boundaries, or by deciding to use more complex while loops rather than a for. A few responses tried to answer the question by repeating essentially the same structure of nested loops three times in cascade, to compute separately the new values in VALUE, COL and ROWC: this was done by introducing a high number of temporary variable. Temporary variables within the body of some loops were not declared nor instantiated in advance.

(d) Some candidates did not attempt this. Those who did usually got it right or obtained partial marks.

(e) Mostly answered correctly.

(f) (i) Mostly answered correctly. Some attempted it but went off-course, and were probably running out of time.

(ii) Only a few responses were fully correct and received full marks. Many responses focused only on the case when R is not 0 and received partial marks, whenever correct.

Recommendations and guidance for the teaching of future candidates

At times, there is little evidence that the basic or foundational concepts involved are indeed correctly understood so to be able to apply them in simple practical situations: this was particularly visible in some responses in relation to code constructions/analysis/code transformation. These candidates have lost marks for very poor specialist terminology and skipping questions on code.

Please continue to teach, teach well, teach Computer Science from the point of view of Computer Science, and with the accuracy and formality required by a scientific subject, especially at HL.

More attention to detail and a more rigorous use of terminology on data structures and operations is needed in responses: so possibly, this style should be also used more strictly in the classroom.

Standard level paper one

Component grade boundaries

Grade:	1	2	3	4	5	6	7
Mark range:	0 – 16	17 – 32	33 – 39	40 – 43	44 – 48	49 – 52	53 - 70

General comments

Many of the comments in this paper also apply to HL Paper 1.

The range and scope of questions in Nov 2017 Paper 1 is comparable to those of previous sessions; however, for this paper several simplifications were made that delivered altogether a more accessible paper. The simplifications include a reduced amount of questions (and marks) on code construction, a structure that reduces the overall number of sub-questions, a better distribution of marks across Section A and B, with more opportunities to hit high scoring questions in B.

The simplifications include a reduced amount of questions in Section A, most of them not particularly mark-heavy, so to support a better time-management before moving to Section B. Section B itself contains a reduced amount of sub-questions, limiting therefore the phenomenon of fragmentation and follow through errors. The content devoted to analysis and construction of code has been reduced, with the side-effect of limiting the possibilities for the examiner to assess algorithmic and procedural thinking.

Reponses show a broad range of abilities in expressing basic and factual knowledge; in general, lack of detail in descriptive question, lack of attention to detail (accuracy) in describing factual knowledge and in code, and limited or even no use of specialist terminology are the main reasons of loss of marks.

In most cases, a bit of attention and detail in writing, and the use of specialist terminology would have helped reducing the loss of marks, that was mostly generated as follows:

1) Schematic (tabular) style of writing is often insufficient to address the explanatory level expected in Assessment Objective 3 (AO3) questions.

2) Very short sentences (half a line) that are not enough to answer a question that is based on an AO3 command term (explain etc);

3) No use of specialist terminology, or improper use of operations proper of data structure understanding them just as basic English words rather that operations in computer science.

Most responses show some attempts of writing code, but some responses indicate little or no idea of how to write a simple loop, no matter whether the attempt is correct or not: occasionally the impression is that some candidates have not even been taught any code using pseudocode at all.

There appeared to be confusion in attempting to write code, even in some of the high-scoring responses. For example, some candidates decided to construct code on ‘collection’ when the question was evidently giving a fixed size array or a matrix. Deciding the boundaries of loops on given structures is a difficulty for many; distinguishing the index of an array with the value store in there is also a problem. Many responses did not show a correct way of the direction of assignment in pseudocode (the expression on the left is assigned the one on the right of the equation). Some identifiers declared as constant can behave as variables. Still errors with mathematical writing of “<” and “>”.

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

Basic knowledge on terms such as client/server, and bit, and essential features of a computer language, buses, role of primary memory are not clear for some. They may have answered going off-course, suggesting at times of not having understood the question.

Most candidates attempted truth tabling, but many constructed an incorrect table with the correct number of lines, and others a table with a very incorrect number of lines (1, 2 or also 8).

Logical diagram on gates received a very high number of incorrect drawings, including those where binary gates just have one wire in input.

Calculating the number of loops performed by a variable has been challenging for many.

Tracing was attempted by the large majority of candidates, but many did it incorrectly and some might have lost some marks out of imprecision in the output.

Complex code was challenging for the large majority: only a few responses attempted to solve it completely, with some imprecision, whereas most of the responses focused on substituting only the initial or only the final part of the array, without necessarily running out of time.

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

Candidates performed well in the descriptive questions that can easily be answered also relying on common knowledge and basic personal experience in the use of ICT. This include the functioning of a spell-checker, the auto-save feature, application software packages, measure to prevent data-loss or preserve data confidentiality, issues in system change-over, notion of protocol, some ideas on packet switching.

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

1. Many responses did not answer the question at all, and focused on the merits of using computer languages instead.

2. Reasonably well answered, although with a large variety of vocabulary that incurred into repetitions, causing loss of marks.

3. (a)(b) large amount of very poor responses without detail, or incorrect ones. Some incorrect answers indicated client and server in terms of a human customer /shop attendant or subscriber of internet services and ISP. Others understood client / server only in terms of database technology and the answer was expressed only in terms of storage management.

4. Generally OK. Some candidates have decided to write extensively about the social benefits of these adaptable devices. Some candidates did not make reference to input device.

5. Many returned a 4-line table with correct configuration in input, but incorrect calculations. Several responses were containing 1 or 2 or 8 lines.

6. Very poorly attempted, with drawings of any kind, apart a few attempts on the correct route but still incorrect.

7. (a)(b) Only a few got them both correct. Some returned N,N others returned N,1. Many skipped (a) and attempted only (b).

   (c) Mostly incorrect. Many answers do not show enough detail on how to write a program, or the conditions in the loop is insufficient. The question explicitly asks for a single while loop, but many candidates wrote two or more loops, or used unusual construction such as “loop while-for”. It is unclear whether the question was understood in full.

8. (a) Some incorrect answers, including “Control Unit” or too vague ones.

   (b) poor use of terminology in expressing the answer, together with common knowledge and use of very naif metaphors have not provided sufficient evidence of understanding the role of address and data bus. Some candidates have variably spoken about the to ‘instruction bus’ or the MAR and MDR instead.

9. Many responses cannot define what a bit is.

10. Generally answered correctly.

11. Generally, the entire question was OK, for those that attempted it in full.

12. (a) Generally answered correctly.

(b) Good understanding, but sometimes expressed without necessary detail or with repetitions, causing a loss of marks.

(c) Marks lost for repetitions, lack of detail, extensive expansion of describing situations without hitting the technical or specific core of the marking points.

(d) Some candidates skipped it, other returned a generic answer on protocol and standards.

(e) Very simplistic descriptions of packet switching that did not go much beyond what was illustrated in the given figure.

13. (a) Some candidates got it right.

(b) Some got it right or almost right (for example, producing an output in all lines). Some tracing was incorrect, stopping at 3 or 4.

(c) Only a few responses are correct and yet incomplete. The majority display a variety of issues including incorrect loop boundaries, inability of using mod (when it is used), bad loop construction, inability of substitute/instantiating/handling the indexes, or a variety thereof (indexes turning negative, or tests on negative indexes!). Most attempts were incomplete anyway in their writing, just sketches, and used a style of syntax that shows no ability of writing code in a reasoned way, not during this exam at least.

(d) No candidates have written an answer in a style that allows to detect an algorithm, as the question suggested. Typically, the answer is always descriptive about what one can do and, depending on how vague the response is, it is difficult to extract algorithmic thinking.

Recommendations and guidance for the teaching of future candidates

Algorithmic thinking as in the last question is a way to reason in abstract way following the procedural workflow without asking the student to do the finer details of building a program: at least this needs to be enforced and practiced with ease.

Many answers show a very weak technical profile and very limited use of specialist terminology: please enforce correct use of terminology while teaching, and do not use story telling or metaphors. Please use correct terminology and please use pseudocode.

Paper two

Component grade boundaries

HL

Grade:	1	2	3	4	5	6	7
Mark range:	0 – 10	11 – 20	21 – 27	28 – 32	33 – 36	37 – 41	42 - 65

SL

Grade:	1	2	3	4	5	6	7
Mark range:	0 – 8	9 – 16	17 – 20	21 – 23	24 – 27	28 – 30	31 - 45

General comments

As the SL paper is a complete subset of the HL paper, the comments from both papers have been integrated into a single report. The question numbers referred to in the question by question analysis are based on the HL paper.

OOP continued to be the most popular option taken but there was a noticeable rise in the number of Web Science entries. The relatively low number of candidates meant that in each option a large proportion of the total entries were supplied by just two or three schools. Therefore, the comments below will tend to reflect the performance of these schools.

There were insufficient entries in Option B, Modelling and Simulation, to be able to write a meaningful report.

Option A: Databases

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

Topics which require more than just memory recall proved difficult. The process of normalisation and importantly the consequences for databases if this is not carried out are not well understood.

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

Candidates were well-prepared for the definition type questions that appear in this topic.

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

SLHL

1a/b. Candidates had an understanding of both but were not always able to produce clear answers.

1c. This was a very straight-forward question.

1d/e. These were similar type questions, but candidates had more difficult in identifying the elements of a data dictionary. This is possibly a concept that could be explored later on in the course when students have a better appreciation of databases in general. Looking at examples of data dictionaries would clearly help.

2a. Candidates tended to know the difference but not all could clearly express this. They should make use of examples whenever these would help to clarify their responses.

2b. This is not an easy concept to explain but most candidates answered well.

2c. This was answered correctly by most.

2d. There was great confusion with these two terms with surprisingly few students clearly differentiating between the two, Candidates must refrain from using the same terms (validate, verify) in their answers. Again, the use of examples helped.

2e. As in (d), this was not particularly well-answered.

2f. This was an expected question and candidates were well-prepared with many providing actual queries.

3a. Candidates tended to comment on the waste of storage space, which is a consequence of redundancy, with fewer explaining the problems caused by inconsistency which is a more important consequence of redundant data.

3b/c/d. The rest of question 3 was not well answered, showing a lack of understanding of the fundamentals of database construction. Few were able to give 3 characteristics of 1NF. Similarly, not many candidates in (c) were able to explain why a single key would not uniquely identify a record and even fewer were able to confidently discuss whether or not the relations in (d) were in 3NF. Several gave out the definition of 3NF without adequately expanding on it.

HL

4a. This was a comparison question which should be answered by comparing similar aspects of both. Candidates who had been using an OOP language throughout the course would have been at an advantage.

4b/c. Both these questions required an adequate knowledge of the course which tended to be school-specific. It was noticeable that not many candidates were able to place this knowledge in context and give good reasons for the practice of segmentation.

4d. A standard question that was quite well answered.

4e. This question wasn't answered well. Not many were able to clearly show the difference.

4f. In a similar manner to 4c. the application of database theory to actual contexts causes difficulties for many students.

Recommendations and guidance for the teaching of future candidates

Candidates tended to perform well on questions requiring factual recall. However, they were not as comfortable when they were faced with questions that placed the theory in context. This was particularly noticeable with the normalization questions. Teachers should, therefore, consider whether they are providing enough examples where students are forced to explore the consequences cause by a lack of normalization. Again, in the HL course, some of the topics will appear quite abstract unless the students are allowed to explore actual examples.

Option C: Web Science

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

Candidates tended to struggle when faced with connecting theory with particular scenarios.

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

General knowledge was quite good. Candidates responded well to definition type questions.

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

SL/HL

10a. This was generally well-answered.

10b. Few candidates were able to discuss both advantages and disadvantages in enough detail. When faced with this type of question they must should look to carefully explain (and not just identify) examples of each.

10c. This was poorly answered. The question focused on reliability but few candidates seemed to have an in-depth knowledge of how these protocols provide this reliability.

10d. Again, many answers lacked detail. Most knew the basic difference between the two types of compression, but candidates should always take the opportunity to include examples or applications in their responses.

11a. This wasn't well-answered showing the lack of practical experience, perhaps, in experimenting with suitable applications. Although the question didn't explicitly ask it, the question was implicitly asking why server-side would be preferred to client-side.

11b. Many candidates were not quite sure what was being asked here although most would have experienced this type of log-in request in their everyday use of the Internet. Only those who had a deep understanding of the functioning of the Internet were able to answer well.

11c. This was generally well-answered.

11d. The question, perhaps, appeared complicated but was made up of distinct stages which the candidates were asked to identify. The practical aspects of this course should lead candidates to be familiar with this type of coding.

12a. The incorporation of dynamic web pages is generally linked to the use of server-side databases and most candidates were able to base their answers on this.

12b. Most came up with two examples (although not all were clearly a result of dynamic scripting), but not all of these specifically answered the question which was asking for benefits,

12c. The idea of increased number of visits leading to the increased number of links leading in from other (authoritative) sites was recognised by most candidates.

12d There were some vague answers here showing a lack of understanding of how these tags are used.

HL

13a/b/c/d/e. Candidates appeared to be well-prepared for this type of question although not all were able to clearly explain why a tendril is so called.

13f. This is a difficult concept that is rarely answered well. Some managed to gain marks with generic descriptions of the power laws.

14a. Most candidates were able to identify the sensors that would be involved in this scenario but were less clear on how they would be used.

14b. Few were able to relate the use of smartphones with their location facility to the actual scenario. Several did comment on how individuals could be tracked.

Recommendations and guidance for the teaching of future candidates

Web Science covers such a potentially large amount of material that it can be a challenge for teachers to adequately prepare students for these exams. The individual sections in the subject guide need to be expanded wherever possible to give students as wide an experience as possible. The practical course should include aspects of both client-side and server-side scripting to allow familiarity with possible questions on these topics.

Option D: OOP

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

Many struggled with some of the basic concepts of OOP such as encapsulation and inheritance. Recursion, as always, seemed to divide the candidates. The use of methods specific to library classes such as the Linked List class was not well understood by all.

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

The students were generally competent at manipulating objects and even linking them with others.

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

SL/HL

15a. Most candidates realised that you could initialise these variables within the main declaration of the object.

15b. There was some confusion here with many confusing static with final and claiming that a static variable could not be changed.

15c. It was not enough just to give basic examples (e.g. int is used for numbers). Full marks went to those who could explain at a processing level why there are different data types.

15d. This was answered well.

15e. This was straight forward for those who had sufficient experience at coding.

16a/b. These are expected questions and were answered well by the majority. Interestingly, b(ii) was the one that caused some problems.

16 c/d. These were the two main coding questions for the SL section with (c) being a basic linear search and (d) involving the combination of more than one object. The latter proved the more difficult with few students managing to correctly use both the tax and the quantity variables. Most students differentiated well between the use of variables and methods, and the use of accessor methods has now been well-established in most schools.

17a. Schools are getting used to this type of question and candidates are starting to appreciate the way that objects relate to each other. Either the use of correct arrows or the naming of relations was accepted with some students providing both.

17b/c. These are both standard topics but many students had difficulties in providing clear responses. When candidates are given a scenario as in part (c), then they should make use of it. Students also need to realise that there are other benefits of encapsulation other than the prevention of accidental changing of variables.

HL

18a. This question was answered well.

18b. The HL course specifically introduces 2 library classes (Linked List and ArrayList) so schools need to teach the methods that are specifically used by these classes. The use of general pseudocode type methods is unlikely to gain credit. It is noticeable that candidates are reluctant to use while loops together with a Boolean variable. There are many circumstances where this type of loop construction is preferable

19a. Candidates struggled with part (a) which suggested a general lack of understanding with the functioning of recursive procedures.

19b. This was a good test for the HL candidates and rewarded those who could both logically deduce the solution to the problem and combine this with a solid understanding of recursion.

Recommendations and guidance for the teaching of future candidates

As was previously mentioned, a couple of schools dominated the entry for this option. It was clear that the candidates from these schools had been well-prepared, almost certainly by being introduced both to basic algorithmic thinking and to the concepts of OOP programming lower down the school. This is not an option that can be successfully addressed in a short period of time as the understanding required can only come from an extended programme.

Higher level paper three

Component grade boundaries

Grade:	1	2	3	4	5	6	7
Mark range:	0 – 4	5 – 9	10 – 11	12 – 14	15 – 16	17 – 19	20 - 30

General comments

This case study focused on the ways that computer based technology is increasingly being incorporated into mainstream medical diagnoses and treatments.

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

There was a noticeable weakness when candidates were asked to adapt their knowledge to specific contexts. Some candidates tended to off-load what they had learned without specifically answering the question. An example of this was in the very first question where many candidates described an application of bioinformatics.

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

Most candidates showed a reasonable general knowledge, although this was school-specific.

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

1a. The vast majority of candidates understood the concept of bioinformatics and gained the mark for the definition. Several then continued by identifying an actual application instead of providing an example of data that might be used.

1b. Most candidates were familiar with the concept of fuzzy logic and the idea that it involves “degrees of truth”. Not all, however, were able to clearly show the value of this with regard to medical diagnosis.

2a. The approach to any comparison question should be to consider various features of both of the elements being compared <u>at the same time</u>. This can either be carried out using a table or in extended writing.

For example: “Ultrasound makes use of sound-waves whilst CT scans use x-rays” is a suitable response.

Candidates who first described ultrasound and then separately described CT scans ran the risk of a failure to match certain features. Candidates should also realise that credit is unlikely to be given for answers taken directly from the case study document unless further expansion is given.

2b. This is an example of the case study linking with the main computer science course. The difficulties involved in the combination of different computer systems can be considerable, and it was these difficulties that this question was probing. A suitable response would have involved identifying a compatibility issue (e.g. use of different operating systems such as Mac vs Windows) and then providing some expansion that clearly demonstrated the candidate’s understanding.

3. The linking of telemedicine with augmented reality was a combination that was probably not directly addressed in the teaching of this case study. This required the candidates to use their understanding of the two areas in providing an explanation of how they could be successfully combined when performing operations. The aspects that should have been covered were in showing clearly what augmented reality was, how it can be used in surgery and the technical elements that must be present to allow real-time communication to take place with an external expert.

4. Question 4 will always require a detailed knowledge and understanding of the computer science aspects of a described scenario. Most candidates displayed a reasonable general knowledge but many tended to include all aspects of the case study without directly relating them to the specific situation outlined in the question. The fact that the facilities would not be that of a major hospital and that the area covered was quite remote were important factors to consider. The higher mark bands were reached by those students who carefully tailored their answers taking all of these specific factors into consideration.

It was noticeable that candidates are now structuring their response to this question well, including both a brief introduction and a reasoned conclusion.

Recommendations and guidance for the teaching of future candidates

The recommendations are similar to those following previous sessions. The students need to have a reasonable understanding of the computer science addressed in each case study. This can only be achieved by the school providing and enacting a structured plan (covering the 12 months prior to the examinations) that addresses this computer science in context. The fact that the case studies and HL Paper 3 follow a similar format each session implies that the same basic plan can be used each year.

The following elements can be considered:

- Analysis of the case study through the use of mind-maps by groups of students
- The use of the summer vacation for an investigation by the students of the new terminology
- The use of visits to installations or of visits by experts to the school
- The importance of individual research to further and deepen understanding