Diploma Programme

Computer science: Frequently asked questions (first assessment 2027)

International Baccalaureate®
Baccalauréat International
Bachillerato Internacional

Diploma Programme

Computer science: Frequently asked questions (first assessment 2027)

International Baccalaureate®
Baccalauréat International
Bachillerato Internacional

Diploma Programme

Computer science: Frequently asked questions (first assessment 2027)

Published October 2025

Published by the International Baccalaureate Organization, a not-for-profit educational foundation of Rue du Pré-de-la-Bichette 1, 1202 Genève, Switzerland.
Website: ibo.org

IB mission statement

The International Baccalaureate aims to develop inquiring, knowledgeable and caring young people who help to create a better and more peaceful world through intercultural understanding and respect.

To this end the organization works with schools, governments and international organizations to develop challenging programmes of international education and rigorous assessment.

These programmes encourage students across the world to become active, compassionate and lifelong learners who understand that other people, with their differences, can also be right.

IB learner profile

The aim of all IB programmes is to develop internationally minded people who, recognizing their common humanity and shared guardianship of the planet, help to create a better and more peaceful world.

As IB learners we strive to be:

INQUIRERS
We nurture our curiosity, developing skills for inquiry and research. We know how to learn independently and with others. We learn with enthusiasm and sustain our love of learning throughout life.

KNOWLEDGEABLE
We develop and use conceptual understanding, exploring knowledge across a range of disciplines. We engage with issues and ideas that have local and global significance.

THINKERS
We use critical and creative thinking skills to analyse and take responsible action on complex problems. We exercise initiative in making reasoned, ethical decisions.

COMMUNICATORS
We express ourselves confidently and creatively in more than one language and in many ways. We collaborate effectively, listening carefully to the perspectives of other individuals and groups.

PRINCIPLED
We act with integrity and honesty, with a strong sense of fairness and justice, and with respect for the dignity and rights of people everywhere. We take responsibility for our actions and their consequences.

OPEN-MINDED
We critically appreciate our own cultures and personal histories, as well as the values and traditions of others. We seek and evaluate a range of points of view, and we are willing to grow from the experience.

CARING
We show empathy, compassion and respect. We have a commitment to service, and we act to make a positive difference in the lives of others and in the world around us.

RISK-TAKERS
We approach uncertainty with forethought and determination; we work independently and cooperatively to explore new ideas and innovative strategies. We are resourceful and resilient in the face of challenges and change.

BALANCED
We understand the importance of balancing different aspects of our lives—intellectual, physical, and emotional—to achieve well-being for ourselves and others. We recognize our interdependence with other people and with the world in which we live.

REFLECTIVE
We thoughtfully consider the world and our own ideas and experience. We work to understand our strengths and weaknesses in order to support our learning and personal development.

The IB learner profile represents 10 attributes valued by IB World Schools. We believe these attributes, and others like them, can help individuals and groups become responsible members of local, national and global communities.

International Baccalaureate®
Baccalauréat International
Bachillerato Internacional

Teaching the course: Curriculum changes and syllabus content

The syllabus content includes many new items such as machine learning, databases for all students and object-oriented programming (OOP) for standard level (SL) students. How do we cope with so much content?	When going through the new Diploma Programme (DP) computer science syllabus, it's important to understand the purpose and balance of the course. The curriculum is designed to be a well-rounded computer science programme with a focus on computational thinking, algorithmic reasoning and programming skills. The new content introduces topics reflecting current technology and developments in the field: machine learning, databases (for all students) and object-oriented programming. To manage the scope and depth of this content, teachers should carefully read the syllabus to understand the boundaries and expectations for each topic. Not all topics require the same level of detail. The depth of teaching should be guided by the command terms used in each learning statement and the associated assessment objectives. The syllabus acknowledges the rapid pace of change in areas like machine learning and databases. It provides clear guidance on what is included, what is not, and how each topic relates to examinations. It's important to approach this syllabus as a new framework, without referencing the previous version.
The course seems to be more challenging than the previous course. Why is this?	This course feels more challenging for several reasons. It places a strong emphasis on computational thinking as a conceptual foundation. **Programming is a required skill for all students, and it is no longer possible to avoid it by selecting certain options.** Additionally, topics that were more aligned with information technology have been removed to focus more on computer science. The syllabus is designed to equip students with current and relevant skills to meet these new demands. This intentional rigour helps build valuable skills and leads to a more rewarding learning experience. The course remains academically rigorous for all students. Higher level (HL) students are being prepared for university-level computer science studies, while SL students are gaining the skills to apply computing in any academic or professional field.
Machine learning is a complex subject area for students. What depth of understanding will be examined?	While machine learning is a complex area, the focus for SL students is conceptual understanding, practical application and real-world relevance along with ethical considerations. The guide suggests or allocates only 5 teaching hours to machine learning out of a total of 105 hours for SL students; teachers do need to keep that in mind when they teach the topic. HL students are expected to delve deeper, with the total number of hours being 18 out of 195 teaching hours. The machine learning section of the syllabus uses command terms to indicate the expected depth of understanding. For example: • supervised learning is linked to AO3 command terms, which require critical evaluation and justification

	• unsupervised and reinforcement learning are linked to AO2, focusing on explanation and application. This helps teachers and students determine how deeply each topic will be examined.
How can teachers upskill in topics that they are not familiar with?	To support upskilling, the teacher support material (TSM) includes guidance on areas where teachers may wish to focus their development. In addition, the IB Exchange for computer science, expected to launch by the end of 2025, will provide a valuable platform for sharing resources, attending webinars and connecting with other educators teaching the course. Teachers are encouraged to identify the parts of the course they feel least confident with when attending workshops or IB Exchange webinars. This helps ensure that future support is targeted and relevant. Currently, machine learning and databases are viewed as the two most important topics requiring attention.
In the textbooks, some syllabus points appear to have been interpreted broadly and encompass a lot of content. Will all of this be assessed?	**While the textbooks are valuable teaching resources, they do not determine what is assessed in external examinations.** They are developed in collaboration with the IB and reviewed by appointed IB reviewers. Although aligned with the course content, the textbooks explore certain topics in greater depth than required for assessment. This is intentional and designed to support better understanding. For example, a concept like the convolutional neural network (CNN) may be included to illustrate key ideas in machine learning. While students will not be assessed on the full technical details of CNNs, such examples help deepen conceptual understanding and make abstract concepts more accessible.

Teaching the course: Developing course outlines

What support can I find for writing course outlines and creating unit plans?	The TSM provides support in course development and unit planning. Also, note that IB Exchange will provide support for unit plans and lesson ideas. The platform will also host webinars and other activities to support the delivery of the DP computer science curriculum.
What are some recommendations for teaching SL and HL students in the same classroom?	Combined SL and HL classes are common in many schools and can be effectively managed through flexible planning. The subject guide and TSM provide suggestions for course organization and programming that can be adapted for mixed classes. Since SL and HL share a substantial portion of the curriculum, particularly in the first year, teachers can deliver the core material to all students together. Teachers can provide additional activities, projects or breakout sessions for HL-only topics, while SL students work on self-directed projects and practice or engaging activities based on the core material during this time. The TSM also includes unit planning examples to help structure this approach, and the upcoming IB Exchange will offer further support through shared resources and professional discussions.
How should guiding questions and linking questions be used when planning units of work?	**Guiding questions** appear at the start of each subtopic to promote inquiry and engage students with key concepts. They serve as suggestions to support unit planning rather than mandatory elements, and teachers may adapt or replace them depending on how they organize their units. These questions reflect the emphasis on conceptual understanding, helping to define the goals of each unit and ensuring alignment between learning outcomes, activities and assessments. Teachers can also revisit guiding questions at the end of a unit to help students reflect on what they have learned and to evaluate the effectiveness of the unit for future planning. While guiding questions are useful starting points, teachers are free to structure their units differently and create their own questions instead. **Linking questions** are designed to help students connect different parts of the syllabus and deepen their understanding of the material. They are not examinable content and should not be treated as topics, subtopics or learning statements. Rather, they encourage students to apply prior knowledge in new contexts, promoting knowledge transfer and highlighting the conceptual coherence of the course. Linking questions help students see relationships between concepts learned in different contexts and gain a more integrated and meaningful understanding of computer science.
What other support is there for unit planning?	Two textbooks have been developed in cooperation with the IB to ensure quality and alignment: • *Oxford Resources for IB: DP Computer Science Course Book*, published by Oxford University Press

	• *Computer Science for the IB Diploma*, published by Hachette Learning. These textbooks provide structured support for planning units and interpreting the guide, offering relevant examples, exercises and explanations.

Assessment: General questions

Will the textbooks be used to determine the level of understanding required for assessments?	No, the textbooks do not determine what will be assessed. **Assessments are based exclusively on the subject guide**, which serves as the definitive reference for both teaching and examination. The guide specifies the required content and the depth of understanding through command terms, which are used for writing assessment tasks and marking student responses.
Why does the specimen paper use "compare and contrast" even though only "compare" is listed as a command term in the subject guide?	"Compare" is an official command term in the subject guide, and it requires students to focus on similarities only. While "compare and contrast" is not listed as a command term, it can still be used in assessments when both similarities and differences are relevant to the question. In such cases, the phrase builds on the recognized command term and adds clarification, without change to the cognitive demand. Its use in the specimen paper was intentional—to prompt a broader comparison.
Will more specimen papers be provided?	The specimen papers published in February 2025 will be the only ones available before live examinations in May 2027. Once examination sessions are held for the second cohort in May 2028, there will be the specimen papers plus at least three sets of past papers available.

Assessment: Paper 1—case study

Why is the case study released only after 12 months of teaching?	The case study is intentionally released 12 months before the May examination session (18 months before the November session), as outlined in the current syllabus. This will allow students to carry out detailed research prior to the examination and teachers to integrate the case study into their curriculum. In the first year, teachers can use the specimen or a previous case study to help students build analytical and research skills: • understanding the types of skills and knowledge assessed • practising case study analysis and related research • exploring the difference in expectations between SL and HL, where HL students are expected to engage in greater depth, as reflected in the mark allocation.
Is there any guidance on which topics students should focus on in the case study?	Students are expected to investigate problems that relate directly to the case study they are given. The last part of the case study, called "Challenges faced", helps students identify which issues to focus on—and these are different for SL and HL students. Students should base their research on the specific challenges listed in that section. Although the case study goes beyond the syllabus, it is released a year before the examination. This gives teachers flexibility: they can either teach the related topics in year 2 if they have not yet, or use the case study to review topics already taught in year 1.
Is there sufficient time for students to answer the case study questions in paper 1 section B?	Yes, there is sufficient time. Section B is worth 12 marks on paper 1 for SL and 24 marks on paper 1 for HL. Since students have about 1.5 minutes for each mark, SL students should aim to spend around 20 minutes answering case study questions, and HL students about 40 minutes. For SL, the first two questions (a) and (b) in section B are "Outline" questions, which only require brief answers as indicated by the answer space provided. Question (c) requires more depth and explanation, and this can be seen in the rubric in the markscheme. This section has a substantial amount of space to answer the question.
Will the questions in the case study section always have a 2, 2, 4, 4, 12 mark structure for HL, and 2, 4, 6 for SL?	The precise breakdown of marks may differ to some extent in future paper 1 case study questions. However, the basic structure will remain the same. Shorter questions aimed at AO1 and AO2 will make up the initial parts and be worth 50% of the marks for the case study question. The remaining 50% of the marks will be for a longer question part that focuses on AO3.
Can the IB provide a recommended reading list of sources for the case study?	There are no plans at this point in time to produce a reading list. However, forums such as IB Exchange can be good places to share ideas about the relevant reading resources and areas for students to focus their research.

Assessment: Paper 1—format

Will the format of the paper in the May 2027 examination session be identical to the specimen papers? Specifically, will there be no blank answer booklets for paper 1, given that paper 1 includes space for answers? Can students be given extra blank sheets of paper if needed?	Yes, the format of the examinations will be the same as in the specimen papers published in the Programme Resource Centre (PRC). Blank answer booklets for paper 1 are not provided, but extra blank sheets of paper will be available if required.
Will the order of questions follow the order of syllabus content as in the specimen papers?	Paper 1 will only have questions from theme A and the case study; paper 2 will only have questions from theme B. In general, the topics in the examination paper will follow the same order as syllabus content, although there may be some exceptions. The first question in paper 2 for each option will always be a question that does not require code.
The questions in paper 1 are based on a scenario that remains the same for all questions for the different topics in theme A, and then there are questions on the case study. Will this be the case for the actual paper in May 2027?	Not necessarily. However, it is recognized that shifting scenarios midway through a paper can require a mental adjustment, which takes time. Where feasible, a consistent scenario will be maintained throughout.

Assessment: Paper 1—syllabus questions

How many inputs do students need to be able to consider for logic gates, truth tables and Boolean expressions?	Students should be able to handle logic circuits with three inputs when constructing and analysing truth tables. If there are more than three inputs, then students will not be required to create a full truth table and would only be asked for the results of specific input values.
How many inputs will be used in the Karnaugh map?	Students should be prepared to handle Karnaugh maps with three variables.
While simplifying the Boolean expressions, will the student be required to know about De Morgan's law and other Boolean laws?	As stated in the subject guide, students need to be able to use "Karnaugh maps and algebraic simplifications to simplify output expressions". If they are required to use algebraic simplifications such as De Morgan's laws, these laws would be provided in the assessment question. For example: Q1. De Morgan's laws can be used to simplify logical expressions to make circuit designs more efficient. The laws are given below. First law: $\overline{A \bullet B} = \overline{A} + \overline{B}$ (The negation of an AND operation is the same as ORing the negations) Second law: $\overline{A + B} = \overline{A} \bullet \overline{B}$ (The negation of an OR operation is the same as ANDing the negations) Describe how the following Boolean expression can be simplified. [2] $$\overline{X + Y + Z}$$ Markscheme *Award [2] max* *Award [1] for using the second law* *Award [2] for extending it to three variables* Apply De Morgan's second law: $\overline{A + B} = \overline{A} \bullet \overline{B}$ Extending this to three variables: $$\overline{X + Y + Z} = \overline{X} \bullet \overline{Y} \bullet \overline{Z}$$
Can multi-valued dependencies exist in a third normal form (3NF) database?	Students will not be assessed on multi-valued dependencies because they exist in a fourth normal form (4NF) or Boyce–Codd normal form. A maximum of 3NF can be asked.
For SL students, using a single OOP class with a database seems insufficient for the internal assessment (IA). To what extent do SL students need to engage with multiple	OOP in SL is for a single class, but students are expected to understand principles of inheritance and polymorphism (which apply for multiple classes). Teachers should provide SL students with an example of OOP for multiple classes to explain inheritance and polymorphism practically. The SL examination papers will not require students to program more than a single class, but students might be asked to explain concepts relating to more than one class. The IA does not require OOP. There is no

classes in OOP in terms of aggregation and inheritance?	requirement for OOP programming, especially if programming a database using Python and SQL.

Assessment: Paper 2

In the specimen paper for SL, there seems to be required knowledge of abstract data types for question 1(e). Is this a mistake?	The first question on the specimen paper 2 for both SL and HL focuses on algorithmic thinking and does not require students to write any code. Students are expected to interpret information algorithmically, and the markscheme confirms that code is not necessary to answer this question. Specifically, question 1(e) on stacks and queues aligns with SL content from the new guide (B2.2.3 and B2.2.4), with the relevant operations fully defined in the question itself. Topic B2.2 in the guide covers data structures, using the following command terms: • "compare" for static and dynamic data structures (B2.2.1) • "construct" for arrays and arraylists (B2.2.2) • "explain" for stacks and queues (B2.2.3 and B2.2.4). Students are expected to understand stack and queue operations conceptually.
Is knowledge of pseudocode required for paper 2?	Pseudocode is not required for paper 2. Students are not expected to use pseudocode in their answers to questions on paper 2, and pseudocode will not be used in any questions. This does not prevent teachers teaching pseudocode to their students to respond to algorithmic thinking questions. This is at the discretion of the teacher and there are no expectations of pseudocode use in examinations.
Question 1(d) is a "describe" question, and the answer in the markscheme is in pseudocode. Is pseudocode required for this question?	Question 1(d) requires a description of algorithm steps which would be an acceptable answer for a "describe" question. **Pseudocode is not required.** Note that the markscheme for the specimen papers has not been standardized. Actual examination papers are standardized by senior examiners using student responses to update the markscheme. The IB assessment division will produce more guidance for the examinations in 2026, a year before the first assessment in 2027. In these live markschemes, examples of answers that do not use pseudocode will be given.
How are Java and Python used in paper 2?	There are two options in paper 2. • Option A has questions in Java. • Option B has questions in Python. Students should always choose the option for the programming language they have studied. Students must not answer questions from different options. Both options are of equivalent difficulty level. Note that the first questions in each option (for SL and HL) are identical. The other questions in option A and option B are equal in number, and corresponding questions use the same context and assess the same skills. Although Python has built-in functions (such as sort, pop, max and min), certain questions on paper 2 will prohibit their use. This information will be clearly indicated in the examination paper.

Will a list of expected disallowed functions in Python be published?	No, a list of Python functions disallowed in paper 2 will not be published in advance. However, the examination paper will make it clear which functions are disallowed for the relevant questions.
How are errors in syntax treated in the marking of examinations?	There may be some penalization for errors in syntax. Small errors may incur no penalty, and repeated errors of the same type would only be penalized at most once. Code with different types of syntax errors would receive some penalty since this suggests a lack of understanding of a language's rules. The focus of assessment is whether the student meets the requirement of the question, not simply correct syntax. However, syntax errors that impact execution, for example, use of a single "=" instead of "==" and incorrect indentation in Python that alters the code structure, may receive some penalty. Students should take care with syntax when writing code in examinations.
Is it more difficult to answer the questions in Java than in Python?	No, the difficulty level is the same for both programming languages in the examination. Students are assessed on their OOP skills, problem-solving and algorithmic thinking, not on language-specific features. That said, Python is seen as more intuitive and beginner-friendly than Java though both support OOP. Python includes extensive libraries that simplify coding tasks, but some of these may be explicitly disallowed in examinations to ensure fairness and to test algorithmic thinking. The choice of which language to teach depends on the teacher's familiarity and confidence with the language.
For paper 2, will there be a Python examination tool subset (PETS) document similar to the Java examination tool subset (JETS) in previous years?	No, there are no plans to publish a PETS document for the examinations. If a particular Python function library made a question much easier, students will not be allowed to use the library in their answer. The examination question will state clearly any functions that are not allowed to be used.
Will the examinations be online in May 2027?	No. The examinations are expected to be on paper, unless a change is announced. Any changes to this will be flagged well in advance of the examinations being online.
In "B2.4.1 Describe the efficiency of specific algorithms by calculating their Big O notation to analyse their scalability", will students be required to do calculations in the examinations?	The command term is "describe", not "calculate". Students need to be able to describe the nature of the calculations—not do the calculation itself.

Internal assessment: Teaching

How can teachers best communicate expectations to students for the IA? For example, how do we make the markbands transparent and clarify the criterion D requirements (list of techniques)?	The IA criteria follow the computational thinking process, so students are expected to choose a problem that they can solve using this process with the skills acquired in the course. The work submitted for assessment should also demonstrate the type of problems and the nature of solutions, with the marks awarded standardized by a senior moderation team. To help students understand expectations: • use the work submitted for assessment as a marking exercise with students to introduce them to the IA • review the markbands together—each one includes the same strands, but the command terms and requirements vary • discuss how command terms determine the markband a response fits into. For criterion D, students must apply techniques from the syllabus. Since the IA is the same for SL and HL, the minimum expectation is SL-level techniques. The techniques chosen should match the success criteria and support both the functionality and efficiency of the solution.
It was mentioned that students are not expected to go beyond what they have been taught in their IAs. On the other hand, we understand that we should encourage well-qualified students to choose topics related to machine learning. Also, some students may choose technologies that are not covered in class, for example creating dynamic web pages or mobile apps. Is this correct?	Students are encouraged to use what they have learned in the course to complete their IA. The work should not require technical abilities beyond the students' capabilities. There may be students who, due to other experiences in programming, wish to use skills that they have acquired beyond the curriculum. However, in the interest of student welfare, it is not advised that students learn large amounts of new material to complete their IAs. In terms of "encouraging well-qualified students to choose topics related to machine learning", this is not a statement seen in IB documentation. There may be opportunities to explore machine learning for some students in their IA, but this would be for highly motivated students with personal interests and, most likely, prior experience.
The current definitions for the IA criterion C and criterion D are very ambiguous and open to many different interpretations. Can the IB clarify what is required?	The exemplars on the PRC will help give examples of the expected responses for criteria C and D. Also, it is recommended that teachers review the TSM which discusses in detail how to interpret these criteria.
Is it correct that only three files are uploaded for the computational solution?	Yes, that is correct. • The documentation (PDF) • The video • The appendices (PDF)

	Remember that the documentation and video are used for marking the computational solution, while the appendices is where additional information that is not marked is submitted. Students need to be clear that the appendices must include the full source code for the solution.
How is it possible to include all the testing, given that the word count is limited?	The assessment criteria are looking for evidence of the testing strategy and not necessarily the actual testing itself. Students can include examples of the strategy being applied in the documentation, but they do not need to provide full testing. This could be included in the appendices if it is important to reference in the documentation. The TSM suggests incorporating test tables for a few major algorithms with observed results. Tables are not part of the word count. The exemplars in the PRC help illustrate this.
Can the planning stage remain AGILE?	"Criterion B: Planning" in the guide states that different planning strategies and representations are possible. There is no stipulation that a plan, once decided upon, cannot change during the development of the solution. The TSM advises that "the planning stage of the documentation should not be written after the computational solution is completed, but beforehand". In other words, planning is a design activity that should not be modified after the product is completed.
Is there a document that describes what is and what is **not** counted in the word count?	The guidance in the subject guide states: "The total word count for the documentation must not exceed 2,000 words. This does not include excerpts of code, comments or diagrams. The overall word count must be clearly written on the first page of the document." In terms of diagrams, words that are used to label or identify the diagram or elements in the diagram are not included in the word count. Tables (as used in design or in testing strategy) are typically not counted in the word count. However, extended writing (like descriptions or explanations) in tables will be part of the word count.
Can we have a checklist for the IA to show complex techniques?	The IA criteria do not refer to "complex techniques" specifically. Complexity is referenced in the introduction to the IA: "The problem chosen should require a software solution of sufficient complexity to be commensurate with the level of this DP computer science course. It should also require sufficient innovation for the student to demonstrate their organizational skills, algorithmic thinking and ability to code their algorithms." Complexity is presented in terms of the problem to be solved with computing commensurate with the course. Overly simplistic tasks usually require minimal to no coding by the candidate. These can include: • static websites • underdeveloped databases, which only store and display data • classroom exercises in coding (basic add/edit/delete/save) without persistence (only primary memory) • websites or databases created from templates • programs created from tutorials • prototypes of a potential solution instead of an actual solution. If students propose only using overly simplistic tasks, teachers should guide them to explore more complex solutions. Students are encouraged

	to use what they have learned in class, and to avoid developing techniques beyond the course.

Internal assessment: Assessment

Should teachers put the comments and marks on the documentation document?	Yes. It is a good idea to annotate the documentation and indicate where evidence for each of the strands in a specific criterion is found. A teacher comment and mark sheet can be included at the beginning of the documentation. This helps assessors understand what teachers used as evidence for a specific markband on each criterion, and also why a mark in that markband was awarded.
How do we use the markbands and award marks?	Awarding marks involves a two-stage process. 1. Identify the appropriate markband by evaluating each strand (bullet point) of the criterion and selecting the descriptor that best fits the student's work. Use a best-fit approach to determine which overall markband applies for the student work. 2. Select the mark within the band based on how well the work meets the descriptors.
Is the video marked?	The video is not marked directly on its own, but it does play a role in providing evidence for criterion D. In particular, functionality can be demonstrated in the video by showing the solution. It also gives examples of the testing strategy. The testing strategy will be graded from the documentation and so the video does not need to show comprehensive testing. Therefore, all work should come with a video and the absence of a video will negatively impact the assessment.
Should students only use Java or Python in the IA?	Students are most likely to use the language they have studied in the course. However, there is no restriction on using a different programming language as long as the code listing has been provided in a readable format (such as PDF) and the language does not trivialize the construction of the solution with minimal or no coding.
In terms of criterion E, if there are no issues with the final product, does that mean the student has no improvement to suggest? Are recommendations for additional functionality valid improvements?	Students should still provide suggestions for improvements even if there are no issues in the final product, as these should address any problems that come out during testing. Examples of suggested improvements include: • expanding the success criteria to include a wider range of inputs into the product • refining efficiency or effectiveness in the functionality of the product • considering some change in requirements that the testing did not capture. It is important that students justify improvements in terms of how they benefit the product meaningfully and their feasibility. Trivial improvements based on poor success criteria or contrived improvements for basic applications are discouraged and will not be assessed positively.

Internal assessment: Moderation

Are the expectations for SL and HL IA the same?	Yes. SL and HL are graded by the same criteria and at the same level. The IB moderation of the computational solution does not distinguish between SL and HL work.
Can non-assessors receive the same document (clarifying the expectations for each criterion) to mark and moderate the IAs?	Yes. The clarification of the expectations for the criteria is included in the subject report after each examination session. A document containing the clarification will also be published on the PRC and will also be used by assessors.
Will the assessors be mindful of the topics learned and skills acquired by the students in the current course when moderating the IAs? We often see a big gap between what we teach and the level of coding required for developing the IAs.	Yes, this issue is acknowledged. The new course is designed to reduce this gap in expectations between the assessments. Students are not expected to go beyond the scope of the syllabus. Since SL and HL students are assessed on the same IA criteria, SL programming should be sufficient. Students should select a problem that is complex enough so they can showcase computational thinking required at the level of the course. Note that moderation is not based on specific topics but rather on how well students apply the computational thinking process—like decomposition, pattern recognition, algorithm design, coding, debugging and testing. Assessors evaluate how clearly students demonstrate and use their skills and understanding through their project.
The criterion D top markband has "**constructs** a fully functional product" and "**constructs** a product that uses appropriate techniques to implement the algorithms". If this is seen for a problem that does not have sufficient complexity, how do we mark this?	The guide includes the following information about the IA: "The problem chosen should require a software solution of sufficient complexity to be commensurate with the level of this DP computer science course. It should also require sufficient innovation for the student to demonstrate their organizational skills, algorithmic thinking and ability to code their algorithms." There is a positive assumption in the criteria that this condition of sufficient complexity has been observed. Assessed student work demonstrates how problems with different levels of complexity or innovation are marked. It is acknowledged that one of the issues with the current IA is that students feel that they have to produce code of much greater complexity than that seen in the course. The aim of the new subject guide is to avoid this issue and focus on the complexity of a problem rather than the complexity of the techniques.