Internal assessment

Purpose of internal assessment

IA is an integral part of the course and is compulsory for both SL and HL students. The assessment criteria are identical for both SL and HL students.

The IA task for computer science is a computational solution. This component brings together the two main themes of the syllabus in a single task where students can demonstrate their knowledge of, and ability to apply, the computational thinking process.

For the computational solution, students:

- select a topic within computer science
- specify a problem of their own choosing
- create a computational solution to this problem that demonstrates their skills and knowledge of the computational thinking process and of their chosen topic.

The IA task should, as far as possible, be woven into normal classroom learning and teaching and not be a separate activity that is unrelated to other elements of the course.

Students can choose their problem from a wide range of contexts, and it should be of personal interest to them.

During their work on the computational solution, students can demonstrate their knowledge of processes such as decomposition, pattern recognition, algorithmic thinking, writing programs, debugging and testing. This task allows them to do this without the time limitations and other constraints associated with written examinations.

The final product of the computational solution is a document that defines the problem and clearly demonstrates the solution using computational thinking in line with the assessment criteria. This will also be accompanied by a video where the student demonstrates the functionality of their solution and examples of their testing strategy.

Guidance and authenticity

The computational solution submitted for assessment must be the student's own work. However, it is not the intention that students should decide upon a title or topic and be left to work on the IA component without any further support from the teacher. The teacher should play an important role during both the planning stage and the period when the student is working on the internally assessed work.

It is the responsibility of the teacher to ensure that students are familiar with the:

- requirements of the type of work to be internally assessed
- assessment criteria—students must understand that the work submitted for assessment must address these criteria effectively.

Teachers and students must discuss the IA work. Students should be encouraged to initiate discussions with the teacher to obtain advice and information, and must not be penalized for seeking guidance. As part of the learning process, teachers should read and give advice to students on one draft of the work. The teacher should provide oral or written advice on how the work could be improved, but not edit the draft. The next version handed to the teacher must be the final version for submission.

It is the responsibility of teachers to ensure that all students understand the basic meaning and significance of concepts that relate to academic integrity, especially authenticity and intellectual property. Teachers must ensure that all student work for assessment is prepared according to the requirements and must

explain clearly to students that the IA work must be entirely their own. Where collaboration between students is permitted, it must be clear to all students what the difference is between collaboration and collusion.

All work submitted to the IB for moderation or assessment must be authenticated by a teacher, and must not include any known instances of suspected or confirmed malpractice. Each student must confirm that the work is their authentic work and constitutes the final version of that work. Once a student has officially submitted the final version of the work, it cannot be retracted. The requirement to confirm the authenticity of work applies to the work of all students, not just the sample work that will be submitted to the IB for the purpose of moderation.

For further details, refer to the IB publications *Academic integrity policy*, *Diploma Programme: From principles into practice* and the relevant general regulations in *Diploma Programme Assessment procedures* (updated annually).

Authenticity may be checked by discussion with the student on the content of the work, and scrutiny of the following.

- The student's initial proposal
- The first draft of the written work
- The references cited
- The style of writing compared with work known to be that of the student
- The analysis of the work by a web-based plagiarism detection service such as turnitin.com

The same piece of work cannot be submitted to meet the requirements of both the IA and the EE.

Time allocation

IA is an integral part of the computer science course, contributing 30% to the final assessment in the SL course and 20% in the HL course.

It is recommended that 35 hours of teaching time should be allocated to the work. This should include:

- time for the teacher to explain to students the requirements of IA
- class time for students to work on the IA component and ask questions
- time for consultation between the teacher and each student
- time to review and monitor progress, and to check authenticity.

Requirements and recommendations

Teachers and students will need to discuss issues relating to the problem the student wants to solve, strategies for developing a solution, and methods for testing and evaluating their solution. Students should be encouraged to use skills of inquiry and research to initiate discussions with the teacher to obtain advice and information, and will not be penalized for seeking support.

Ethical guidelines

Given the nature of the IA task, students must take into account ethical problems and implications concerning undertaking research and developing the solution. For example, they should ensure the confidentiality and security of data. Wherever possible, original data should be used or collected by the student.

The following guidelines must be applied.

- Consent must be obtained from people who will be involved in the development of the computational solution before any investigation begins.
- Written consent must be obtained from the owner of any existing system that will be used as part of the IA—for example, when implementing a security analysis protocol on an existing system.
- All data collected must be stored securely to maintain confidentiality.

- Data collected can only be used for the computational solution. It must not be used for any other purpose without explicit permission.

Health and safety guidelines

Schools are advised to follow local best practice in health and safety for research linked to the development of the computational solution. Each school is ultimately responsible for the health and safety of students.

Using assessment criteria for internal assessment

For IA, a number of assessment criteria have been identified. Each assessment criterion has level descriptors describing specific achievement levels, together with an appropriate range of marks. The level descriptors concentrate on positive achievement, although for the lower levels failure to achieve may be included in the description.

Teachers must judge the internally assessed work at SL and at HL against the criteria using the level descriptors.

- The same assessment criteria are provided for SL and HL.
- The aim is to find, for each criterion, the descriptor that conveys most accurately the level attained by the student, using the best-fit model. A best-fit approach means that compensation should be made when a piece of work matches different aspects of a criterion at different levels. The mark awarded should be one that most fairly reflects the balance of achievement against the criterion. It is not necessary for every single aspect of a level descriptor to be met for that mark to be awarded.
- When assessing a student's work, teachers should read the level descriptors for each criterion until they reach a descriptor that most appropriately describes the level of the work being assessed. If a piece of work seems to fall between two descriptors, both descriptors should be read again and the one that more appropriately describes the student's work should be chosen.
- Where there are two or more marks available within a level, teachers should award the upper marks if the student's work demonstrates the qualities described to a great extent; the work may be close to achieving marks in the level above. Teachers should award the lower marks if the student's work demonstrates the qualities described to a lesser extent; the work may be close to achieving marks in the level below.
- Only whole numbers should be recorded; partial marks (fractions and decimals) are not acceptable.
- Teachers should not think in terms of a pass or fail boundary but should concentrate on identifying the appropriate descriptor for each assessment criterion.
- The highest level descriptors do not imply faultless performance but should be achievable by a student. Teachers should not hesitate to use the extremes if they are appropriate descriptions of the work being assessed.
- A student who attains a high achievement level in relation to one criterion will not necessarily attain high achievement levels in relation to the other criteria. Similarly, a student who attains a low achievement level for one criterion will not necessarily attain low achievement levels for the other criteria. Teachers should not assume that the overall assessment of the students will produce any particular distribution of marks.
- It is recommended that the assessment criteria be made available to students.

Internal assessment details—SL and HL

Duration: 35 hours

Weighting: SL 30%, HL 20%

Introduction

The IA task requires the student to identify a problem of their own choosing and develop a software solution for it, using the computational thinking process.

The solution is assessed using five criteria.

- Criterion A: Problem specification
- Criterion B: Planning
- Criterion C: System overview
- Criterion D: Development
- Criterion E: Evaluation

Key terms

Solution: This refers to the documentation and video submitted by the student for the IA.

Product: This refers to the completed software only.

Choice of problem

In identifying a problem, the student can select to apply to the problem any topic in computer science that interests them. It does not have to be directly related to the specified themes in the syllabus.

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.

More examples and detail of the features of complex and innovative solutions are given in the *Computer science teacher support material*.

The nature of the solution

All solutions must be coded and can take a number of forms. These can include:

- creating a new system, such as an OOP program, interactive web-based application using a database, computer game, mobile application, simulation, stand-alone application, web-based application
- adding functionality to an existing system, such as connecting a webpage(s) to a database, writing a function for Moodle, writing a plug-in, or developing a stand-alone application.

Whichever problem and form of solution the student chooses, it is essential that they explicitly demonstrate and document their algorithmic thinking skills. Products that take existing templates that show no evidence of modification in their structure, design or functionality are not appropriate. Examples of **inappropriate** products include:

- the development of a programming product using only copied code
- the development of a website (product) using a web-based template that predetermines its structure and layout
- the use of exemplar products or templates provided with software, (e.g. the Northwind database in Microsoft Access)
- a copied computer game without major modifications to the code that have been properly documented
- a product that does not meet the ethical requirements outlined in the "Requirements and recommendations" section of this publication
- a computer/mobile application created using a builder/wizard/drag-and-drop tool without the need for code development.

Requirements

The IA submission consists of three types of files.

- Documentation

- Video
- Appendices

Documentation

The documentation must be submitted as a single PDF file.

The documentation must include five separate sections, one for each of the five criteria.

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.

More detail on the nature of the code excerpts, and related comments, are given in the *Computer science teacher support material*.

Video

The purpose of the video is to provide evidence of the functionality of the product and to give examples of the testing strategy. The video must be submitted as a separate file. It must also:

- be no longer than five minutes
- be in a commonly used format such as mp4, .avi or .wmv
- demonstrate the full functionality of the product
- demonstrate examples of the testing strategy used in the development of the product.

More support on the nature and making of the video can be found in the *Computer science teacher support material*.

Appendices

Appendices must be submitted as a single PDF file. The appendices must include the full source code and any other resources developed by the student that are referred to in the documentation.

The appendices are **not** used as evidence for awarding marks, and examiners are not required to read the appendices. However, solutions that do not include an appendix with the full source code cannot be awarded full marks for the techniques demonstrated in criterion D.

Task descriptors and criteria—SL and HL

The computer science IA focuses on using the computational thinking processes and the skills of algorithmic thinking and programming to solve a problem chosen by the student. The computational solution is about problem-solving.

The assessment criteria

Criteria A, B, C and E are process-oriented. The criteria both guide and assess how the IA task is carried out and allow common assessment criteria to be applied to different types of products.

Criterion D is an assessment of the final product and assesses the student's understanding of the concepts involved in its development. The ability to carry out a testing strategy on the final product is a key element in this criterion. Examples from the testing strategy should be demonstrated in the video and will be used as part of the evidence for this criterion.

Criterion A: Problem specification (4 marks)

The problem specification is the starting point of the solution and must be used as a basis for the development of the product.

- The student should have the necessary technical skills, access to appropriate hardware and software, and availability of relevant data to address the problem.
- The success criteria identified in the problem specification (criterion A) will be used in the planning (criterion B), in the development (criterion D) and in the evaluation (criterion E).

The recommended word count for this criterion is **300 words**.

Marks	Level descriptor
0	The response does not reach a standard described by the descriptors below.
1–2	The response: • **outlines** a problem scenario • **states** limited success criteria • **outlines** the nature of the solution in a computational context.
3–4	The response: • **describes** the problem scenario in terms of its measurable solution requirements • **states** appropriate success criteria • **explains** the choice of computational context for the solution.

Problem specification clarifications
The **problem scenario** is a clear description of the problem including its measurable solution requirements. The description should relate directly to the problem, whether this be in the world around us, other fields of knowledge, or a current issue in computing. **Success criteria** are measurable outcomes derived from the solution requirements that indicate the successful development of the product. The **computational context** is the specific area of computing that is selected to be used in the solution.

Criterion B: Planning (4 marks)

The planning of the product must be consistent with the problem specification in criterion A.

- This criterion assesses how the problem scenario has been decomposed into component parts.
- The plan should address the requirements of the solution in terms of the success criteria, and include a proposed chronology for the steps involved in planning, designing, developing, testing and evaluating the solution.
- A plan can be presented in different forms, but popular diagram formats such as GANTT and AGILE charts can effectively support the planning process.
- The plan may include any relevant research, such as the use of existing code libraries.

The recommended word count for this criterion is **150 words**.

Marks	Level descriptor
0	The response does not reach a standard described by the descriptors below.
1–2	The response: • **constructs** a partial decomposition of the problem scenario • **constructs** a plan that addresses some of the success criteria of the solution.
3–4	The response: • **constructs** a reasonable decomposition of the problem scenario • **constructs** a plan that addresses the success criteria of the solution.

Planning clarifications
Decomposition is the breaking down of the problem scenario identified in criterion A into smaller, more manageable sub-problems or components. The decomposition can be effectively constructed using diagrams. A **reasonable decomposition** breaks the problem down into essential components that support the construction of a plan.

Criterion C: System overview (6 marks)

The system overview of the product must be consistent with the problem specification in criterion A and the planning in criterion B.

- The system overview should include a system model with the key components, their relationships, the rules governing their interaction, and the algorithms required by these components and the user interface.
- The system overview should have the clarity to enable a third party to recreate the product.
- The system model will provide the information for a viable testing strategy.

The recommended word count for this criterion is **150 words**.

Marks	Level descriptor
0	The response does not reach a standard described by the descriptors below.
1–2	The response: • **outlines** a limited system model • **identifies** algorithms for the components of the system model • **identifies** a testing strategy for at least one success criterion.
3–4	The response: • **constructs** a system model that is not complete • **constructs** algorithms for the components of the system model that lead to partial functionality of the product • **outlines** a testing strategy that aligns with at least three success criteria.
5–6	The response: • **constructs** a complete system model • **constructs** algorithms for the components of the system model that enable the product to perform • **describes** a testing strategy that aligns with the success criteria.

System overview clarifications
A **system model** consists of diagrams that include the components of the system and how they are connected. The system model will include the design of the user interface. A **complete system model** does not include the algorithms for each of the components. **Algorithms** can be presented in different forms, including natural language, flow charts or pseudocode, and should address the individual components of the system model. The **testing strategy** refers to a systematic approach for evaluating whether the computational solution works as intended. The testing strategy should ensure that code functions correctly and handles

System overview clarifications
unexpected or incorrect inputs. This can be represented effectively in a table with proposed test data and expected outcomes.

Criterion D: Development (12 marks)

The development of the product must be consistent with the problem specification in criterion A, the planning in criterion B and the system overview developed in criterion C.

- The video must provide evidence of the functionality and give examples of the testing of the product.
- The development of the solution must justify the structure of the product, why the structure is appropriate, and demonstrate the techniques used to develop the product based on the algorithms constructed in criterion C. These techniques may include loops, data structures, existing libraries and the integration of software tools.
- The testing strategy must include testing for correctness, reliability and efficiency. The testing must be described and justified in the documentation with supporting examples seen in the video.

The recommended word count for this criterion is **1,000 words**.

Marks	Level descriptor
0	The response does not reach a standard described by the descriptors below.
1–3	The response: • **constructs** a product with very limited functionality • **constructs** a product using no appropriate techniques to implement the algorithms • **states** the choices made to implement the algorithms • **states** the testing strategy used.
4–6	The response: • **constructs** a product that has limited functionality • **constructs** a product using at least one appropriate technique to implement the algorithms • **outlines** the choices made to implement the algorithms • **states** the effectiveness of the testing strategy.
7–9	The response: • **constructs** a product that has partial functionality • **constructs** a product that uses some appropriate techniques to implement the algorithms • **explains** the choices made to implement the algorithms • **describes** the effectiveness of the testing strategy.
10–12	The response: • **constructs** a fully functional product • **constructs** a product that uses appropriate techniques to implement the algorithms • **evaluates** the choices made to implement the algorithms • **justifies** the effectiveness of the testing strategy.

Development clarifications
Implementation and coding of the algorithms: Techniques in the criteria refer to the process of programming algorithms using code. The documentation must highlight key elements of code that are important for the efficient functioning of the algorithms. Any code presented in the solution must include relevant comments, be consistent and be readable. Code excerpts included in the documentation must be referenced to the full source code submitted as an appendix. The video must demonstrate the **functionality** of the product. The deployment of the **testing** strategy and its effectiveness must be described in the documentation, with examples of the testing seen in the video.

Criterion E: Evaluation (4 marks)

The evaluation of the product must be consistent with the problem specification and success criteria in criterion A.

The recommended word count for this criterion is **400 words**.

Marks	Level descriptor
0	The response does not reach a standard described by the descriptors below.
1–2	The response: • **states** the extent to which the success criteria were met • **describes** improvements to the product.
3–4	The response: • **evaluates** the extent to which the success criteria were met • **justifies** improvements to the product.