International Baccalaureate®
Baccalauréat International
Bachillerato Internacional

Diploma Programme

Computer science guide

First examinations 2014

International Baccalaureate®
Baccalauréat International
Bachillerato Internacional

Diploma Programme

Computer science guide

First examinations 2014



Diploma Programme
Computer science guide

Published January 2012

Published on behalf of the International Baccalaureate Organization, a not-for-profit educational foundation of 15 Route des Morillons, 1218 Le Grand-Saconnex, Geneva, Switzerland by the

International Baccalaureate Organization (UK) Ltd
Peterson House, Malthouse Avenue, Cardiff Gate
Cardiff, Wales CF23 8GL
United Kingdom
Phone: +44 29 2054 7777
Fax: +44 29 2054 7778
Website: www.ibo.org

© International Baccalaureate Organization 2012

The International Baccalaureate Organization (known as the IB) offers three high-quality and challenging educational programmes for a worldwide community of schools, aiming to create a better, more peaceful world. This publication is one of a range of materials produced to support these programmes.

The IB may use a variety of sources in its work and checks information to verify accuracy and authenticity, particularly when using community-based knowledge sources such as Wikipedia. The IB respects the principles of intellectual property and makes strenuous efforts to identify and obtain permission before publication from rights holders of all copyright material used. The IB is grateful for permissions received for material used in this publication and will be pleased to correct any errors or omissions at the earliest opportunity.

All rights reserved. No part of this publication may be reproduced, stored in a retrieval system, or transmitted, in any form or by any means, without the prior written permission of the IB, or as expressly permitted by law or by the IB's own rules and policy. See http://www.ibo.org/copyright.

IB merchandise and publications can be purchased through the IB store at http://store.ibo.org. General ordering queries should be directed to the Sales and Marketing Department in Cardiff.

Phone: +44 29 2054 7746
Fax: +44 29 2054 7779
Email: sales@ibo.org

International Baccalaureate, **Baccalauréat International** and **Bachillerato Internacional** are registered trademarks of the International Baccalaureate Organization.

Printed in the United Kingdom by Antony Rowe Ltd, Chippenham, Wiltshire

4061

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.

IB learners strive to be:

Inquirers	They develop their natural curiosity. They acquire the skills necessary to conduct inquiry and research and show independence in learning. They actively enjoy learning and this love of learning will be sustained throughout their lives.
Knowledgeable	They explore concepts, ideas and issues that have local and global significance. In so doing, they acquire in-depth knowledge and develop understanding across a broad and balanced range of disciplines.
Thinkers	They exercise initiative in applying thinking skills critically and creatively to recognize and approach complex problems, and make reasoned, ethical decisions.
Communicators	They understand and express ideas and information confidently and creatively in more than one language and in a variety of modes of communication. They work effectively and willingly in collaboration with others.
Principled	They act with integrity and honesty, with a strong sense of fairness, justice and respect for the dignity of the individual, groups and communities. They take responsibility for their own actions and the consequences that accompany them.
Open-minded	They understand and appreciate their own cultures and personal histories, and are open to the perspectives, values and traditions of other individuals and communities. They are accustomed to seeking and evaluating a range of points of view, and are willing to grow from the experience.
Caring	They show empathy, compassion and respect towards the needs and feelings of others. They have a personal commitment to service, and act to make a positive difference to the lives of others and to the environment.
Risk-takers	They approach unfamiliar situations and uncertainty with courage and forethought, and have the independence of spirit to explore new roles, ideas and strategies. They are brave and articulate in defending their beliefs.
Balanced	They understand the importance of intellectual, physical and emotional balance to achieve personal well-being for themselves and others.
Reflective	They give thoughtful consideration to their own learning and experience. They are able to assess and understand their strengths and limitations in order to support their learning and personal development.

© International Baccalaureate Organization 2007

Contents

Introduction 1

Purpose of this document 1
The Diploma Programme 2
Nature of the subject 4
Aims 8
Assessment objectives 9
Assessment objectives in practice 10

Syllabus 11

Syllabus outline 11
Approaches to the teaching and learning of computer science 12
Syllabus content 17

Assessment 65

Assessment in the Diploma Programme 65
Assessment outline—SL 67
Assessment outline—HL 68
External assessment 70
Internal assessment 75
Summary of the group 4 project 87

Appendices 93

Glossary of command terms 93
Further reading 95

Purpose of this document

This publication is intended to guide the planning, teaching and assessment of the subject in schools. Subject teachers are the primary audience, although it is expected that teachers will use the guide to inform students and parents about the subject.

This guide can be found on the subject page of the online curriculum centre (OCC) at http://occ.ibo.org, a password-protected IB website designed to support International Baccalaureate (IB) teachers. It can also be purchased from the IB store at http://store.ibo.org.

Additional resources

Additional publications such as teacher support materials, subject reports, internal assessment guidance and grade descriptors can also be found on the OCC. Specimen and past examination papers as well as markschemes can be purchased from the IB store.

Teachers are encouraged to check the OCC for additional resources created or used by other teachers. Teachers can provide details of useful resources, for example, websites, books, videos, journals or teaching ideas.

Acknowledgment

The IB wishes to thank the educators and associated schools for generously contributing time and resources to the production of this guide.

First examinations 2014

The Diploma Programme

The Diploma Programme is a rigorous pre-university course of study designed for students in the 16 to 19 age range. It is a broad-based two-year course that aims to encourage students to be knowledgeable and inquiring, but also caring and compassionate. There is a strong emphasis on encouraging students to develop intercultural understanding, open-mindedness, and the attitudes necessary for them to respect and evaluate a range of points of view.

The Diploma Programme hexagon

The course is presented as six academic areas enclosing a central core. It encourages the concurrent study of a broad range of academic areas. Students study: two modern languages (or a modern language and a classical language); a humanities or social science subject; an experimental science; mathematics; one of the creative arts. It is this comprehensive range of subjects that makes the Diploma Programme a demanding course of study designed to prepare students effectively for university entrance. In each of the academic areas students have flexibility in making their choices, which means they can choose subjects that particularly interest them and that they may wish to study further at university.

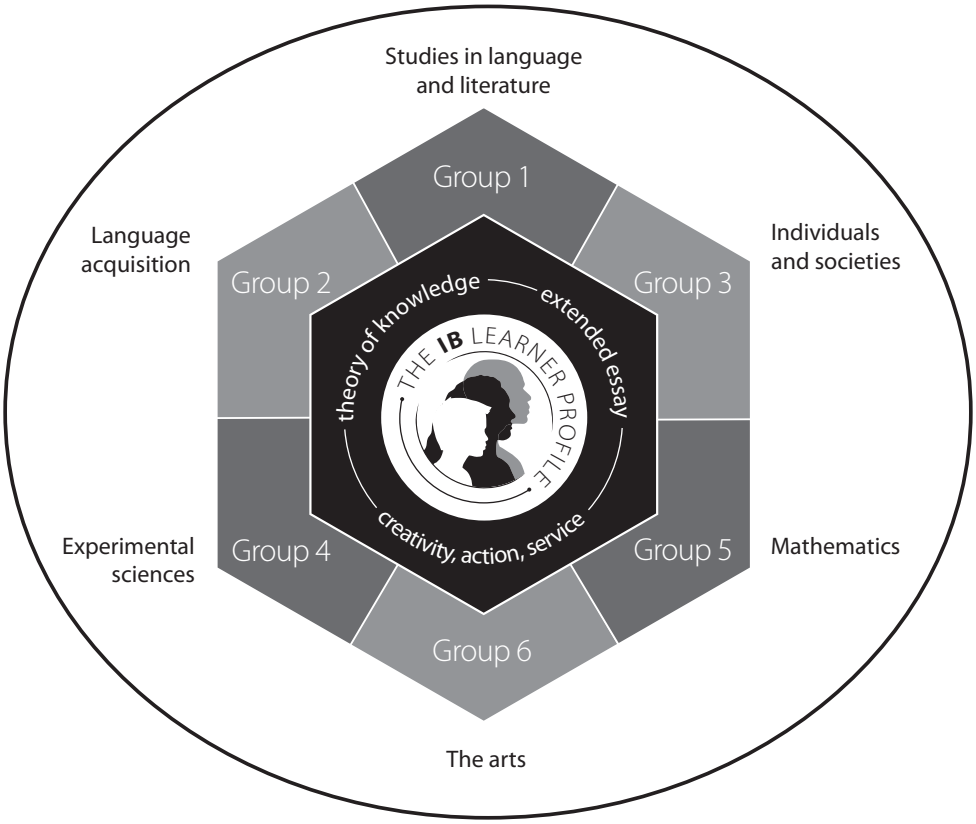


Figure 1
Diploma Programme model

Choosing the right combination

Students are required to choose one subject from each of the six academic areas, although they can choose a second subject from groups 1 to 5 instead of a group 6 subject. Normally, three subjects (and not more than four) are taken at higher level (HL), and the others are taken at standard level (SL). The IB recommends 240 teaching hours for HL subjects and 150 hours for SL. Subjects at HL are studied in greater depth and breadth than at SL.

At both levels, many skills are developed, especially those of critical thinking and analysis. At the end of the course, students' abilities are measured by means of external assessment. Many subjects contain some element of coursework assessed by teachers. The course is available for examinations in English, French and Spanish.

The core of the hexagon

All Diploma Programme students participate in the three course requirements that make up the core of the hexagon. Reflection on all these activities is a principle that lies at the heart of the thinking behind the Diploma Programme.

The theory of knowledge course encourages students to think about the nature of knowledge, to reflect on the process of learning in all the subjects they study as part of their Diploma Programme course, and to make connections across the academic areas. The extended essay, a substantial piece of writing of up to 4,000 words, enables students to investigate a topic of special interest that they have chosen themselves. It also encourages them to develop the skills of independent research that will be expected at university. Creativity, action, service involves students in experiential learning through a range of artistic, sporting, physical and service activities.

The IB mission statement and the IB learner profile

The Diploma Programme aims to develop in students the knowledge, skills and attitudes they will need to fulfill the aims of the IB, as expressed in the organization's mission statement and the learner profile. Teaching and learning in the Diploma Programme represent the reality in daily practice of the organization's educational philosophy.

Nature of the subject

Computer science requires an understanding of the fundamental concepts of computational thinking as well as knowledge of how computers and other digital devices operate.

The Diploma Programme computer science course is engaging, accessible, inspiring and rigorous. It has the following characteristics.

- draws on a wide spectrum of knowledge
- enables and empowers innovation, exploration and the acquisition of further knowledge
- interacts with and influences cultures, society and how individuals and societies behave
- raises ethical issues
- is underpinned by computational thinking.

Computational thinking involves the ability to:

- think procedurally, logically, concurrently, abstractly, recursively and think ahead
- utilize an experimental and inquiry-based approach to problem-solving
- develop algorithms and express them clearly
- appreciate how theoretical and practical limitations affect the extent to which problems can be solved computationally.

During the course the student will develop computational solutions. This will involve the ability to:

- identify a problem or unanswered question
- design, prototype and test a proposed solution
- liaise with clients to evaluate the success of the proposed solution and make recommendations for future developments.

Computer science has links with subjects outside of group 4, notably information technology in a global society (ITGS), but it should be noted that there are clear differences between the subjects.

Computer science and the international dimension

Computer science itself is an international endeavour—the exchange of information and ideas across national boundaries has been essential to the progress of the subject. This exchange is not a new phenomenon but it has accelerated in recent times with the development of information and communication technologies.

The development of solutions may be at a local, national or global scale and lies at the heart of the subject. Therefore teachers of computer science should study a range of examples from different geographical locations as well as at different scales.

Developments such as open source software and the emergence of social networking epitomize the global nature of the subject. Internet forums exist that welcome ideas and solutions developed from computer scientists from all continents in driving forward developments to different software types. These developments have revolutionized the way that people, and in particular the young, interact.

On a practical level, the group 4 project (which all science students must undertake) mirrors the work of computer scientists by encouraging collaboration between schools across the regions.

Distinction between SL and HL

While the skills and activities of computer science are common to students at both SL and HL, students at HL are required to study additional topics in the core, a case study and also extension material of a more demanding nature in the option chosen. The distinction between SL and HL is therefore one of both breadth and depth.

Additionally, the HL course has 240 hours devoted to teaching, compared with 150 hours for the SL course.

Students at SL and HL in computer science study a common core consisting of:

- four topics (system fundamentals; computer organization; networks; and computational thinking, problem-solving and programming)
- one option (chosen from databases; modelling and simulation; web science; or object-oriented programming)
- one piece of internally assessed work, which includes a computational solution.

The HL course has three additional elements:

- three further topics (abstract data structures; resource management; control)
- additional and more demanding content for the option selected
- an additional externally assessed component based on a pre-seen case study of an organization or scenario; this requires students to research various aspects of the subject—which may include new technical concepts and additional subject content—in greater depth.

Prior learning

Past experience shows that students will be able to study computer science at SL successfully with no background in, or previous knowledge of, computer science. Their approach to study, characterized by specific IB learner profile attributes—inquirers, thinkers and communicators—will be significant here. Students who have undertaken the IB Middle Years Programme (MYP) or studied a similar course prior to commencing the IB Diploma Programme would also be well prepared.

The study of computer science at HL demands a higher level of problem-solving skills and the ability to understand and manipulate abstract concepts. Although no previous knowledge of computer science is required, some exposure to programming is desirable.

Links to the Middle Years Programme

Students who have undertaken the MYP sciences, technology and mathematics courses will be well prepared for group 4 subjects. The MYP science objectives and assessment criteria A–F are aligned with the group 4 objectives and internal assessment criteria, and allow for a smooth transition from the MYP to the Diploma Programme. In particular, the "One world" objective in MYP sciences is further developed in group 4 computer science with the increased emphasis on aim 8—that is, to "raise awareness of the moral, ethical, social, economic and environmental implications of using science and technology". There are specific references to aim 8 implications in assessment statements and teacher's notes in the syllabus details sections in all group 4 guides.

The IB technology continuum

The MYP technology course builds on experiences of inquiry that students have gained in their time in the IB Primary Years Programme (PYP). PYP teaching and learning experiences challenge students to be curious, ask questions, explore and interact with the environment physically, socially and intellectually to construct meaning and refine their understanding. Even when there is no technology component in the PYP, the use of structured inquiry is a precursor to the problem-solving and inquiry-based approach of MYP technology.

The MYP technology course develops skills linked to the design cycle, which provides the model of thinking and the strategy used to help students investigate problems and design, plan, create and evaluate the product. Students continuing on to the IB Diploma Programme will have experienced the use of the design cycle and will have developed critical-thinking and design skills, which they will be able to apply and extend in computer science.

As in the MYP technology course, a **product** can be defined as the solution that students have generated independently. This means that the Diploma Programme computer science course requires students to become actively involved in, and to focus on, the whole design process as well as the final product, building on the skills acquired during the MYP technology course.

In order to complete the IB Diploma Programme computer science internal assessment successfully, students are expected to create a solution to a specific problem using the design cycle, which extends the range of skills developed in MYP technology.

Computer science and theory of knowledge

There is no one scientific method of gaining knowledge or of finding explanations for the behaviour of the natural world. Computer science works through a variety of approaches to produce these explanations, but they all rely on data from observations and have a common underpinning rigour, whether using inductive or deductive reasoning. The explanation may be in the form of a theory, sometimes requiring a model that contains elements not directly observable. Producing these theories often requires an imaginative, creative leap. Where such a predictive theoretical model is not possible, the explanation may consist of identifying a correlation between a factor and an outcome. This correlation may then give rise to a causal mechanism that can be experimentally tested, leading to an improved explanation. All these explanations require an understanding of the limitations of data, and the extent and limitations of our knowledge. Computer science requires freedom of thought and open-mindedness, and an essential part of the process of science is the way the international computer science community shares ideas through academic papers, conferences and open forums. The syllabus details sections in the group 4 guides give references in teacher's notes to appropriate topics where theory of knowledge can be addressed.

During the course in computer science a number of issues will arise that highlight the relationships between theory of knowledge and computer science. Some of the questions that could be considered during the course are identified in the following list.

- What is the difference between data, information, knowledge and wisdom? To what extent can computers store and impart data, information, knowledge and wisdom?
- Computational thinking includes: procedure, logic, pre-planning (thinking ahead), concurrency, abstraction and recursion. To what extent are these ways of thinking distinct? To what extent can knowledge in different areas (mathematics, ethics, and so on) be analysed in these ways?
- It has been said that human memory is more like an improvised performance than a movie on a DVD. What does this mean? How does human memory differ from computer memory?
- How does a computer language differ from a natural language?

- What are the differences between representing numbers in denary and in binary? In binary, 1 + 1 = 10. Does this tell us anything about the nature of mathematical truth?
- What are the challenges of creating a computer model of some aspect of the world?
- A chess machine can beat the top human chess players. Does a machine therefore "know" how to play chess?
- To what extent does computational thinking challenge conventional concepts of reasoning?
- How do we know if other humans feel emotions? Can a machine ever feel an emotion? How would we know?
- Was Akio Morita correct when he claimed that "You can be totally rational with a machine. But if you work with people, sometimes logic has to take a back seat to understanding"?
- Does information and communication technology, like deduction, simply allow the knower to arrange existing knowledge in a different way, without adding anything, or is this arrangement itself knowledge in some sense?
- What did Sydney Harris mean when he said that "The real danger is not that computers will begin to think like men, but that men will begin to think like computers"? Was he right, or was this statement based on a misunderstanding of either men or computers?
- What do we mean by "holistic" and "reductionist" approaches to knowledge? What are the strengths and weaknesses of each approach?
- To what extent is it possible to capture the richness of concepts such as "intelligence" or "judgment" via a reductionist approach?
- If we attach a camera or microphone to a computer, it can receive data from the world. Does this mean that a computer can "perceive the world"? To what extent might human perception be a similar process?

Aims

Subject aims

Diploma Programme computer science students should become aware of how computer scientists work and communicate with each other and with other stakeholders in the successful development and implementation of IT solutions. While the methodology used to solve problems in computer science may take a wide variety of forms, the group 4 computer science course emphasizes the need for both a theoretical and practical approach.

It is in this context that the Diploma Programme computer science course should aim to:

1. provide opportunities for study and creativity within a global context that will stimulate and challenge students developing the skills necessary for independent and lifelong learning
2. provide a body of knowledge, methods and techniques that characterize computer science
3. enable students to apply and use a body of knowledge, methods and techniques that characterize computer science
4. demonstrate initiative in applying thinking skills critically to identify and resolve complex problems
5. engender an awareness of the need for, and the value of, effective collaboration and communication in resolving complex problems
6. develop logical and critical thinking as well as experimental, investigative and problem-solving skills
7. develop and apply the students' information and communication technology skills in the study of computer science to communicate information confidently and effectively
8. raise awareness of the moral, ethical, social, economic and environmental implications of using science and technology
9. develop an appreciation of the possibilities and limitations associated with continued developments in IT systems and computer science
10. encourage an understanding of the relationships between scientific disciplines and the overarching nature of the scientific method.

Assessment objectives

The objectives for all group 4 subjects reflect those parts of the aims that will be assessed. Wherever appropriate, the assessment will draw upon a range of contexts and identify the social, moral and economic effects of science and technology.

It is the intention of the Diploma Programme computer science course that students achieve the following objectives.

1. Know and understand:
    a. relevant facts and concepts
    b. appropriate methods and techniques
    c. computer science terminology
    d. methods of presenting information.
2. Apply and use:
    a. relevant facts and concepts
    b. relevant design methods and techniques
    c. terminology to communicate effectively
    d. appropriate communication methods to present information.
3. Construct, analyse, evaluate and formulate:
    a. success criteria, solution specifications including task outlines, designs and test plans
    b. appropriate techniques within a specified solution.
4. Demonstrate the personal skills of cooperation and perseverance as well as appropriate technical skills for effective problem-solving in developing a specified product.

Assessment objectives in practice

The following tables show the approximate percentage weighting in a typical examination session for each of the assessment objectives across each of the components. This may differ from the allocation of time devoted to each of the assessment objectives in class.

Standard level

Assessment objective	Paper 1	Paper 2	Internal assessment	Overall
1. Demonstrating knowledge and understanding	24	13	9	46
2. Applying and using	13	7	8	28
3. Constructing, analysing, evaluating and formulating	8	5	4	17
4. Using skills	n/a	n/a	9	9
Component weighting	**45%**	**25%**	**30%**	**100%**

Higher level

Assessment objective	Paper 1	Paper 2	Paper 3	Internal assessment	Overall
1. Demonstrating knowledge and understanding	21	10	9	6	46
2. Applying and using	12	6	7	5	30
3. Constructing, analysing, evaluating and formulating	7	4	4	3	18
4. Using skills	n/a	n/a	n/a	6	6
Component weighting	**40%**	**20%**	**20%**	**20%**	**100%**

Syllabus outline

Syllabus component	Teaching hours	
	SL	HL
Core syllabus content		
SL/HL core The topics that must be studied, including some practical work, are: • Topic 1: System fundamentals (20 hours) • Topic 2: Computer organization (6 hours) • Topic 3: Networks (9 hours) • Topic 4: Computational thinking, problem-solving and programming (45 hours)	80	80
HL extension The topics that must be studied, including some practical work, are: • Topic 5: Abstract data structures (23 hours) • Topic 6: Resource management (8 hours) • Topic 7: Control (14 hours)	---	45
Case study Additional subject content introduced by the annually issued case study	---	30
Option		
SL/HL core	30	30
HL extension Students study one of the following options: Option A: Databases Option B: Modelling and simulation Option C: Web science Option D: Object-oriented programming (OOP)	---	15
Internal assessment		
Solution Practical application of skills through the development of a product and associated documentation	30	30
Group 4 project	10	10
Total teaching hours	**150**	**240**

It is essential that teachers are allowed the prescribed minimum number of teaching hours necessary to meet the requirements of the computer science course. At SL the minimum prescribed number of hours is 150 hours and at HL it is 240 hours.

Approaches to the teaching and learning of computer science

The Diploma Programme computer science course adopts a significantly different approach to many other computer science courses. Teachers who have taught other courses, including previous IB computer science courses, must familiarize themselves with this approach before they commence teaching.

Computer science is seen as an experimental science. Therefore it is expected that teachers use a wide range of practical activities to support the theoretical content.

Computational thinking

Computational thinking is a problem-solving methodology that is applicable across a range of subject disciplines and underpins this course.

The six principles of computational thinking, identified by Jeanette Wing in her article "Computational thinking" are:

- thinking procedurally
- thinking logically
- thinking ahead
- thinking concurrently
- thinking abstractly
- thinking recursively **(HL only)**.

The 10 hours explicitly stated in the SL/HL core indicate the time required to teach these principles within other topics throughout the duration of the course. In the HL extension the time required to teach these principles is not explicitly stated but is included in the overall time allocated. It is expected that teachers will use an iterative method in integrating these principles into the other subject content.

There are a number of possible approaches to the teaching of these principles such as:

- teaching each principle linked to one scenario only
- teaching the principle(s) as threads that are linked to a number of scenarios
- teaching some of the principles as linked to one scenario only and others linked to a number of scenarios.

Delivering the course

There are a number of approaches that may be taken to deliver the computer science course. The syllabus is written in a non-prescriptive manner that allows appropriate scenarios to be selected by the teacher. The scenarios should include real-world problems that will be relevant, contemporary in nature and provide ways for the students to integrate their own experiences within an inquiry-based approach. These scenarios should provide students with an opportunity to decompose a system, gaining an understanding of algorithms that influence the way it functions.

Topics should not be taught in a linear fashion as it is possible for more than one of the sub-topics to be taught together. This is demonstrated in the example class activity below, which focuses on the development of a login interface.

Class activity	Topic in syllabus
Creating a login interface	• Identify the context for which a system is planned (1.1.1) • Suggest various types of testing (1.1.7) • Construct suitable representations to illustrate system requirements (1.2.7) • Describe the purpose of prototypes to demonstrate the proposed solution to the client (1.2.8) • Thinking procedurally, thinking logically, thinking abstractly (4.1) • Connecting computational thinking and program design (4.2) • Use of programming languages (4.3.6–4.3.9) • Some of the 30 hours allocated to the preparation and/or completion of the internal assessment where techniques can be applied appropriately

The use of code is seen as an essential part of testing the algorithms developed. The teacher support material includes suggested practical activities linked to these types of activity that may be used by teachers as a part of their teaching. This list of practical activities is not exhaustive.

The internal assessment should provide an opportunity for the students to showcase their innovative skills while developing a practical solution to either a specified problem or unanswered question. These skills will be evident in the use of complex design and algorithmic principles as well as rigorous testing leading to the development of a functional product.

Algorithmic development

Teachers must be aware of the different levels of abstraction used in developing a solution to a specified problem. This may be illustrated using the following examples.

Example 1

Given the following array

NAMES

[0]	[1]	[2]	[3]	[4]
Robert	Boris	Brad	George	David

and the following algorithm, which is constructed to reverse the contents of array NAMES

```
N = 5 // the number of elements in the array
K = 0 // this is the first index in the array

loop while K < N - 1
      TEMP = NAMES[K]
      NAMES [K] = NAMES [N - K -1]
      NAMES [N - K -1] = TEMP
      K = K + 1
end loop
```

a. trace the algorithm, showing contents of the array after each execution of the loop [2 marks]

b. state the number of times the loop is executed [1 mark]

c. outline why the algorithm does not reverse the contents of the array NAMES, and how this could be corrected. [2 marks]

Example 2

A geography teacher is searching an array, CITYNAMES, containing 100 names of cities and wants to print out the names of the cities beginning with D.

Construct pseudocode to indicate how this may be done:

```
// FirstLetter("CITY") will return the first letter of the word "CITY"
// Elements are stored in an array called CITYNAMES

loop for C from 0 to 99
      if FirstLetter(CITYNAMES[C]) = "D" then
            output CITYNAMES[C]
      endif
end loop
```

Example 3

A geography teacher is searching CITIES, a collection of city names, and wants to print out the names of the cities beginning with D.

Construct pseudocode to indicate how this may be done:

```
// FirstLetter("CITY") will return the first letter of the word "CITY"

loop while CITIES.HasNextItem()
      NAME = CITIES.getNext()
      if FirstLetter(NAME) = "D"
            output NAME
      end if
end loop
```

Students are not expected to construct code for SL paper 1, HL paper 1 and HL paper 3 beyond the level of pseudocode. For SL paper 2 and HL paper 2, the use of code will depend on the option.

Options

The introduction of options allows computer science teachers to focus on an area of interest or specialism. Below is some information that may assist teachers in selecting the most appropriate option.

Additional information linked to the options

Databases

Students are expected to use database software to support the theoretical concepts within the syllabus.

There is no prescribed database development environment. Appropriate database development environments may include Access, Open Office Base, MySQL and FileMaker.

Modelling and simulation

Students are expected to develop models and/or simulations that support the concepts in the syllabus.

There is no prescribed development environment. Appropriate resources may include Scratch, Alice, Java, Python, Visual Basic (.Net), spreadsheet software and JavaScript.

Web science

Students are expected to use web authoring and other appropriate software to support the theoretical concepts within the syllabus.

There is no prescribed software. Appropriate resources may include WYSIWYG ("what you see is what you get") web page editors, HTML editors, CMS, PHP, JavaScript, Web 2.0 and CGI scripting.

OOP

For the external assessment of this option Java is the prescribed language. However, any appropriate object-oriented language may be used for teaching this option.

Further information on all the above is available in the teacher support material.

Further guidance

Primary and secondary information

Primary information is that information the student collects in discussion with the client. It can include qualitative and quantitative information. Secondary information is information that has already been compiled in a variety of written and electronic forms.

The IB learner profile

The computer science syllabus is closely linked to the IB learner profile. The computer science syllabus allows students to address all the attributes of the IB learner profile.

For each attribute of the learner profile, a number of examples selected from the computer science syllabus are given below.

Learner profile attribute	Computer science syllabus
Inquirers	Content: SL/HL core, HL extension, case study. Decomposing systems to find underpinning algorithms. Solution: Investigation into appropriate scenario.
Knowledgeable	Content: SL/HL core, HL extension, case study. Solution: Justify appropriate techniques in developing the product or modifying an existing product.
Thinkers	Content: HL extension, formulating strategic plans in case study. Solution: Develop an extensible product so it can be maintained by others.
Communicators	Content: SL/HL core, making links to theory of knowledge. Solution: Liaise appropriately with client and/or adviser to develop product that meets their requirements.
Principled	Content: SL/HL core, HL extension, case study. Proposing solutions that are ethical and legal. Solution: Test the product to ensure it is error free and secure, protect any sensitive data.

Learner profile attribute	Computer science syllabus
Open-minded	Content: SL/HL core, HL extension, case study. Respecting differing cultures and opinions of others. Solution: Evaluate potential sources of information in terms of reliability, bias, relevance and accuracy.
Caring	Content: SL/HL core, HL extension, case study. Considering the opinions of different stakeholders when reaching a decision. Solution: Liaise with client, reach consensus in development of product.
Risk-takers	Content: Case study, formulate strategic plans. Solution: Make and justify decisions about which techniques to use in developing the most appropriate product.
Balanced	Content: SL/HL core, HL extension in analysis and judgment-type questions. Solution: Collection of data and the subsequent analysis and synthesis of the information to determine the most appropriate product.
Reflective	Content: Case study, reflecting on possible decisions related to a strategic decision. Solution: Evaluate methodologies used to develop product to recommend future improvements.

The online curriculum centre and workshops

All teachers of the computer science course are strongly encouraged to access the OCC at regular intervals. The main computer science subject page contains essential documents including the guide, teacher support materials, specimen papers and subject reports and important subject updates. All teachers can post inquiries, share examples of good practice, ask for advice and access exemplar materials. The OCC includes frequently asked questions and updates on resources. The computer science forum on the OCC is a forum for computer science teachers to talk to each other.

From time to time special events are held to give teachers an opportunity to participate in activities such as marking work with the opportunity to gain feedback from senior examiners.

Teachers are also encouraged to participate in face-to-face workshops and online workshops. These provide an opportunity to discuss particular aspects of the subject and engage in workshop activities that support the teaching of computer science.

Syllabus content

SL/HL core—80 hours

Due to the rapidly changing nature of computer science it may be necessary to add a list of additional technical terms for SL and HL students. This will be posted on the OCC in addition to the annually issued case study. The list of terms will apply to examinations from the May session two years after the date of posting.

For example, in May 2014 the following information will be posted on the OCC:

- the annually issued case study for May 2015 and November 2015
- additional vocabulary linked to syllabus for first examinations May 2016.

Further information that builds on the information in this guide will be also posted on the OCC under the **General documents** tab:

- approved notation sheet, to be taken into external examinations
- Java Examination Tool Subset (JETS) documentation for students taking the OOP option
- zip file to provide the structure for the internal assessment.

The following command terms will be used in examination questions. It is important that students are familiar with the definitions of the command terms (see Appendices).

The objective level associated with each command term (see objective column) indicates the depth of treatment for a given assessment statement.

- **Assessment objective 1**: classify, define, draw, label, list, state
- **Assessment objective 2**: annotate, apply, calculate, describe, design, distinguish, estimate, identify, outline, present, trace
- **Assessment objective 3:** analyse, comment, compare, compare and contrast, construct, contrast, deduce, demonstrate, derive, determine, discuss, evaluate, examine, explain, formulate, interpret, investigate, justify, predict, sketch, suggest, to what extent

Teacher's notes, which are included alongside some assessment statements, provide further guidance to teachers. In these, links to other topics are shown (**LINK**), and they may also suggest ideas for the promotion of specific subject aims (**AIM**), theory of knowledge (**TOK**), Middle Years Programme (**MYP**), social/ethical issues (**S/E**) and the international dimension (**INT**).

Topic 1—System fundamentals (20 hours)

1.1 Systems in organizations (10 hours)

	Assessment statement	Obj	Teacher's notes
Planning and system installation			
1.1.1	Identify the context for which a new system is planned.	2	The extent and limitations of a new system should be appreciated. Organizational issues related to the installation of new systems such as user roles, underlying technologies.
1.1.2	Describe the need for change management.	2	Students should understand there are a number of factors that need to be managed to ensure change is successful. **S/E** The way that change is managed can have significant effects on employers and employees.
1.1.3	Outline compatibility issues resulting from situations including legacy systems or business mergers.	2	**INT**, **S/E** When organizations interact, particularly on an international basis, there may be issues of software compatibility and language differences.
1.1.4	Compare the implementation of systems using a client's hardware with hosting systems remotely.	3	The benefits and drawbacks of SaaS (Software-as-a-Service) should be considered. **S/E**, **INT**, **AIM 8** The remote host may be in a different time zone and this can have significant effects on end-users.
1.1.5	Evaluate alternative installation processes.	3	Students should be aware of the methods of implementation/ conversion. Parallel running, pilot running, direct changeover and phased conversion. **S/E** Training issues may require organizations to restructure their workforce.
1.1.6	Discuss problems that may arise as a part of data migration.	3	**INT** These include incompatible file formats, data structures, validation rules, incomplete data transfer and international conventions on dates, currencies and character sets.

	Assessment statement	Obj	Teacher's notes
1.1.7	Suggest various types of testing.	3	The crucial importance of testing at all stages of implementation should be emphasized, with the stages clearly defined. Types of testing can include: user acceptance testing, debugging, beta testing. Students should be aware that there are programs that can test other programs, thereby automating parts of the testing process and reducing costs. **S/E** Inadequate testing can reduce employee productivity and lead to end-user dissatisfaction.
User focus			
1.1.8	Describe the importance of user documentation.	2	**S/E** The quality of user documentation can affect the rate of implementation of the new system.
1.1.9	Evaluate different methods of providing user documentation.	3	Examples should include methods such as: help files, online support and printed manuals. **S/E** The quality of user documentation can affect the rate of implementation of the new system.
1.1.10	Evaluate different methods of delivering user training.	3	Examples should include self-instruction, formal classes, remote/online training. **S/E** The quality of the delivery of user training can affect the rate of implementation of the new system.
System backup			
1.1.11	Identify a range of causes of data loss.	2	Causes include malicious activities and natural disasters. **S/E** Malicious activity may be a result of activities by employees within the organization or intruders.
1.1.12	Outline the consequences of data loss in a specified situation.	2	**S/E** Loss of medical records, cancellation of a hotel reservation without the knowledge of the traveller.

	Assessment statement	Obj	Teacher's notes
1.1.13	Describe a range of methods that can be used to prevent data loss.	2	These should include failover systems, redundancy, removable media, offsite/online storage.
Software deployment			
1.1.14	Describe strategies for managing releases and updates.	2	Students should be aware of a variety of ways in which updates and patches are made available and deployed. This includes automatic updates received on a regular basis online. **S/E**, **INT** Performance issues related to the inability to install updates may hinder end-users and reduce compatibility between systems in geographically diverse locations.

1.2 System design basics (10 hours)

	Assessment statement	Obj	Teacher's notes
Components of a computer system			
1.2.1	Define the terms: hardware, software, peripheral, network, human resources.	1	
1.2.2	Describe the roles that a computer can take in a networked world.	2	Roles include client, server, email server, DNS server, router and firewall.
1.2.3	Discuss the social and ethical issues associated with a networked world.	3	**AIM 8**, **AIM 9** Develop an appreciation of the social and ethical issues associated with continued developments in computer systems.
System design and analysis			
1.2.4	Identify the relevant stakeholders when planning a new system.	2	**S/E** The role of the end-user must be considered when planning a new system. Who is a relevant stakeholder? **TOK** Utilitarianism, the greatest good for the greatest number. The means justify the ends.

	Assessment statement	Obj	Teacher's notes
1.2.5	Describe methods of obtaining requirements from stakeholders.	2	Including surveys, interviews, direct observations. **AIM 5** The need for effective collaboration to obtain appropriate information from stakeholders. **S/E** The question of privacy for stakeholders.
1.2.6	Describe appropriate techniques for gathering the information needed to arrive at a workable solution.	2	Examining current systems, competing products, organizational capabilities, literature searches. **S/E** Intellectual property.
1.2.7	Construct suitable representations to illustrate system requirements.	3	Examples include: system flow charts, data flow diagrams, structure chart. UML is not required. **LINK** Flow chart symbols, flow charts and pseudocode.
1.2.8	Describe the purpose of prototypes to demonstrate the proposed system to the client.	2	**AIM 5** The need to effectively collaborate to gather appropriate information to resolve complex problems. **AIM 6** To develop logical and critical thinking to develop proposed systems.
1.2.9	Discuss the importance of iteration during the design process.	3	**MYP** Design cycle.
1.2.10	Explain the possible consequences of failing to involve the end-user in the design process.	3	**S/E** The failure to involve the end-user may lead to software that is not suitable for its intended use, which may have adverse effects on user productivity. **AIM 5** The need for effective collaboration and communication between the client, developer and end-user.
1.2.11	Discuss the social and ethical issues associated with the introduction of new IT systems.	3	**AIM 8**, **AIM 9** Develop an appreciation of the social and ethical issues associated with continued developments in specified computer systems.

	Assessment statement	Obj	Teacher's notes
Human interaction with the system			
1.2.12	Define the term usability.	1	**S/E** This includes ergonomics and accessibility.
1.2.13	Identify a range of usability problems with commonly used digital devices.	2	**S/E** Students should be aware of usability issues in a range of devices including PCs, digital cameras, cell phones, games consoles, MP3 players and other commonly used digital devices.
1.2.14	Identify methods that can be used to improve the accessibility of systems.	2	**S/E** Examples include touch screen, voice recognition, text-to-speech, Braille keyboard.
1.2.15	Identify a range of usability problems that can occur in a system.	2	**S/E** These should be related to the systems. Systems include ticketing, online payroll, scheduling, voice recognition, systems that provide feedback.
1.2.16	Discuss the moral, ethical, social, economic and environmental implications of the interaction between humans and machines.	3	**AIM 8** Raise awareness of the moral, ethical, social, economic and environmental implications of using science and technology.

Topic 2—Computer organization (6 hours)

2.1 Computer organization (6 hours)

	Assessment statement	Obj	Teacher's notes
Computer architecture			
2.1.1	Outline the architecture of the central processing unit (CPU) and the functions of the arithmetic logic unit (ALU) and the control unit (CU) and the registers within the CPU.	2	Students should be able to reproduce a block diagram showing the relationship between the elements of the CPU, input and output and storage. The memory address register (MAR) and memory data register (MDR) are the only ones that need to be included.
2.1.2	Describe primary memory.	2	Distinguish between random access memory (RAM) and read-only memory (ROM), and their use in primary memory.

	Assessment statement	Obj	Teacher's notes
2.1.3	Explain the use of cache memory.	3	Students should be able to explain the effect of cache memory in speeding up the system as well as being able to explain how it is used.
2.1.4	Explain the machine instruction cycle.	3	This should include the role of data bus and address bus.
Secondary memory			
2.1.5	Identify the need for persistent storage.	2	Persistent storage is needed to store data in a non-volatile device during and after the running of a program. **LINK** Consequences of data loss. **TOK** If there are no consequences of data loss, why is it stored. **TOK** There is no such thing as persistent storage. **AIM 9** An appreciation of the issues related to both the ever increasing amount of data and a need to retain it.
Operating systems and application systems			
2.1.6	Describe the main functions of an operating system.	2	This is confined to a single-user operating system. Technical details are not needed. For example, memory management should be described but how this is handled in a multitasking environment is not expected.
2.1.7	Outline the use of a range of application software.	2	Application software should include word processors, spreadsheets, database management systems, email, web browsers, computer-aided design (CAD) and graphic processing software.

	Assessment statement	Obj	Teacher's notes
2.1.8	Identify common features of applications.	2	Including toolbars, menus, dialogue boxes, graphical user interface (GUI) components. Students should understand that some features are provided by the application software and some by the operating system. **S/E** This improves usability for a wide range of users. **AIM 9** An appreciation of the improvements associated with developments in application software.
Binary representation			
2.1.9	Define the terms: bit, byte, binary, denary/decimal, hexadecimal.	1	
2.1.10	Outline the way in which data is represented in the computer.	2	To include strings, integers, characters and colours. This should include considering the space taken by data, for instance the relation between the hexadecimal representation of colours and the number of colours available. **TOK**, **INT** Does binary represent an example of a lingua franca? **S/E**, **INT** Comparing the number of characters needed in the Latin alphabet with those in Arabic and Asian languages to understand the need for Unicode.
Simple logic gates			
2.1.11	Define the Boolean operators: AND, OR, NOT, NAND, NOR and XOR.	1	**LINK** Introduction to programming, approved notation sheet.
2.1.12	Construct truth tables using the above operators.	3	For example, Maria won't go to school if it is cold and raining or she has not done her homework. Not more than three inputs are used. **LINK** Thinking logically. **TOK** Reason as a way of knowing.

	Assessment statement	Obj	Teacher's notes
2.1.13	Construct a logic diagram using AND, OR, NOT, NAND, NOR and XOR gates.	3	Problems will be limited to an output dependent on no more than three inputs. The gate should be written as a circle with the name of the gate inside it. For example: OR **LINK** Thinking logically, connecting computational thinking and program design, introduction to programming.

Topic 3—Networks (9 hours)

3.1 Networks (9 hours)

	Assessment statement	Obj	Teacher's notes
Network fundamentals			
3.1.1	Identify different types of networks.	2	Examples include local area network (LAN), virtual local area network (VLAN), wide area network (WAN), storage area network (SAN), wireless local area network (WLAN), internet, extranet, virtual private network (VPN), personal area network (PAN), peer-to-peer (P2P). **S/E**, **INT** Globalization has been accelerated by the technical advances linked to network development.
3.1.2	Outline the importance of standards in the construction of networks.	2	**INT** Standards enable compatibility through a common "language" internationally.
3.1.3	Describe how communication over networks is broken down into different layers.	2	Awareness of the OSI seven layer model is required, but an understanding of the functioning of each layer is not.
3.1.4	Identify the technologies required to provide a VPN.	2	
3.1.5	Evaluate the use of a VPN.	3	**S/E**, **AIM 9** The use of a VPN has led to changes in working patterns.

	Assessment statement	Obj	Teacher's notes
Data transmission			
3.1.6	Define the terms: protocol, data packet.	1	
3.1.7	Explain why protocols are necessary.	3	Including data integrity, flow control, deadlock, congestion, error checking.
3.1.8	Explain why the speed of data transmission across a network can vary.	3	
3.1.9	Explain why compression of data is often necessary when transmitting across a network.	3	**S/E**, **INT** Compression has enabled information to be disseminated more rapidly.
3.1.10	Outline the characteristics of different transmission media.	2	Characteristics include: speed, reliability, cost and security. Transmission media include: metal conductor, fibre optic, wireless.
3.1.11	Explain how data is transmitted by packet switching.	3	
Wireless networking			
3.1.12	Outline the advantages and disadvantages of wireless networks.	2	**S/E** wireless networks have led to changes in working patterns, social activities and raised health issues.
3.1.13	Describe the hardware and software components of a wireless network.	2	
3.1.14	Describe the characteristics of wireless networks.	2	Include: WiFi; Worldwide Interoperability for Microwave Access (WiMAX); 3G mobile; future networks. **S/E**, **INT** Connectivity between different locations.
3.1.15	Describe the different methods of network security.	2	Include encryption types, userID, trusted media access control (MAC) addresses. **S/E** Wireless networks have led to concerns about the security of the user's data.
3.1.16	Evaluate the advantages and disadvantages of each method of network security.	3	

Topic 4—Computational thinking, problem-solving and programming (45 hours)

4.1 General principles (10 hours)

This should not be taught as a separate topic but must be incorporated and connected to all sections—especially flow charts, pseudocode and programming in the SL/HL core and abstract data structures (HL extension). It is essential that these elements are not addressed in isolation—they have to be approached as a whole.

The basic ideas and their application should be illustrated with non-computer examples. Each basic idea should then be practised in specific algorithmic contexts using concepts drawn from flow charts, pseudocode and programming. The teacher support material illustrates examples such as the home/locker/knapsack for thinking ahead.

	Assessment statement	Obj	Teacher's notes
Thinking procedurally			
4.1.1	Identify the procedure appropriate to solving a problem.	2	This includes identifying the steps and putting them in the correct order. Such as recipes, block-arrow-block-arrow. **LINK** Connecting computational thinking and program design, introduction to programming.
4.1.2	Evaluate whether the order in which activities are undertaken will result in the required outcome.	3	Links to problems presented to the student in other areas of the syllabus. **LINK** Thinking ahead, thinking concurrently. Connecting computational thinking and program design, introduction to programming. **MYP** Technology, step-by-step instructions.
4.1.3	Explain the role of sub-procedures in solving a problem.	3	Constructing procedures that can then be referred to by their identifier. **LINK** Abstraction, connecting computational thinking and program design, introduction to programming.
Thinking logically			
4.1.4	Identify when decision-making is required in a specified situation.	2	Links to procedural thinking—alternative procedures. **TOK** Reasoning as a form of decision-making. **LINK** Connecting computational thinking and program design, introduction to programming.

	Assessment statement	Obj	Teacher's notes
4.1.5	Identify the decisions required for the solution to a specified problem.	2	Different actions are taken based on conditions. **LINK** Connecting computational thinking and program design, introduction to programming. **AIM 4** Applying thinking skills to identify and resolve a specified complex problem.
4.1.6	Identify the condition associated with a given decision in a specified problem.	2	Testing conditions, iteration. Identifying and constructing the conditions—AND, OR, NOT relationships—Boolean tests. **LINK** Connecting computational thinking and program design, introduction to programming.
4.1.7	Explain the relationship between the decisions and conditions of a system.	3	IF … THEN … ELSE **LINK** Connecting computational thinking and program design, introduction to programming.
4.1.8	Deduce logical rules for real-world situations.	3	**LINK** Connecting computational thinking and program design, introduction to programming.
Thinking ahead			
4.1.9	Identify the inputs and outputs required in a solution.	2	
4.1.10	Identify pre-planning in a suggested problem and solution.	2	Gantt charts. Pre-ordering. Pre-heating an oven. Home/locker/knapsack. Caching/pre-fetching. Building libraries of pre-formed elements for future use. **LINK** Thinking procedurally, thinking concurrently. Connecting computational thinking and program design, introduction to programming.
4.1.11	Explain the need for pre-conditions when executing an algorithm.	3	
4.1.12	Outline the pre- and post-conditions to a specified problem.	2	For example, cooking a dish for a meal. All ingredients available before starting to cook. A place to eat the food.

	Assessment statement	Obj	Teacher's notes
4.1.13	Identify exceptions that need to be considered in a specified problem solution.	2	For example, identify the pre-conditions for calculating the end-of-year bonus when not all employees have worked for the company for the whole year. **LINK** Connecting computational thinking and program design, introduction to programming.
Thinking concurrently			
4.1.14	Identify the parts of a solution that could be implemented concurrently.	2	Could include computer systems or real-life situations. **LINK** Thinking ahead, thinking procedurally. Connecting computational thinking and program design, introduction to programming.
4.1.15	Describe how concurrent processing can be used to solve a problem.	2	For example, building a house, production lines, division of labour. Students will not be expected to construct a flow chart or pseudocode related to concurrent processing.
4.1.16	Evaluate the decision to use concurrent processing in solving a problem.	3	**LINK** Thinking ahead, thinking procedurally. Connecting computational thinking and program design, introduction to programming.
Thinking abstractly			
4.1.17	Identify examples of abstraction.	2	Selecting the pieces of information that are relevant to solving the problem. **LINK** Thinking ahead.
4.1.18	Explain why abstraction is required in the derivation of computational solutions for a specified situation.	3	Students should be aware of the concept of objects, for example, the use of collections as objects in the design of algorithms. **LINK** • Databases: tables, queries • Modelling and simulation: an abstraction of reality • OOP: classes, sub-classes • Web science: distributed applications

	Assessment statement	Obj	Teacher's notes
4.1.19	Construct an abstraction from a specified situation.	3	There is no need to use code. Levels of abstraction through successive decomposition. A school can be decomposed into faculties. A faculty can be decomposed into departments. **LINK** Thinking ahead, thinking procedurally. Connecting computational thinking and program design, introduction to programming.
4.1.20	Distinguish between a real-world entity and its abstraction.	2	**TOK** The map as an abstraction of the territory.

4.2 Connecting computational thinking and program design (22 hours)

The focus of this topic is how an understanding of programming languages enhances the students' understanding of computational thinking and provides them with opportunities for practical, hands-on experience of applying computational thinking to practical outcomes.

In externally assessed components questions will be presented using flow charts and/or pseudocode as outlined in the approved notation sheet. Answers will only be required in pseudocode.

Students must be given the opportunity to convert algorithms into working code that is executed and tested.

Working code will not be assessed in the externally assessed components.

	Assessment statement	Obj	Teacher's notes
4.2.1	Describe the characteristics of standard algorithms on linear arrays.	2	These are: sequential search, binary search, bubble sort, selection sort.
4.2.2	Outline the standard operations of collections.	2	These are: addition and retrieval of data.
4.2.3	Discuss an algorithm to solve a specific problem.	3	Students should be expected to discuss the differences between algorithms, including both standard and novel algorithms. For example, discussing the advantages and disadvantages of using a binary search as opposed to a sequential search.

	Assessment statement	Obj	Teacher's notes
4.2.4	Analyse an algorithm presented as a flow chart.	3	Examination questions may involve variables, calculations, simple and nested loops, simple conditionals and multiple or nested conditionals. This would include tracing an algorithm as well as assessing its correctness. Students will not be expected to construct a flow chart to represent an algorithm in an externally assessed component. **MYP** Mathematics: using flow charts to solve problems in real-life contexts, patterns and sequences, logic, algorithms. **MYP** Technology: design cycle (inputs, processes, outputs, feedback, iteration).
4.2.5	Analyse an algorithm presented as pseudocode.	3	Examination questions may involve variables, calculations, simple and nested loops, simple conditionals and multiple or nested conditionals. This would include tracing an algorithm as well as assessing its correctness. **MYP** Mathematics: using flow charts to solve problems in real-life contexts, patterns and sequences, logic, algorithms. **MYP** Technology: design cycle (inputs, processes, outputs, feedback, iteration).
4.2.6	Construct pseudocode to represent an algorithm.	3	**MYP** Mathematics: using flow charts to solve problems in real-life contexts, patterns and sequences, logic, algorithms. **MYP** Technology: design cycle (inputs, processes, outputs, feedback, iteration). **AIM 4** Demonstrate thinking skills to represent a possible solution to a specified complex problem.

	Assessment statement	Obj	Teacher's notes
4.2.7	Suggest suitable algorithms to solve a specific problem.	3	Suitable algorithms may include both standard algorithms and novel algorithms. Suitable may include considerations of efficiency, correctness, reliability and flexibility. Students are expected to suggest algorithms that will actually solve the problem successfully. **LINK** General principles of computational thinking, introduction to programming.
4.2.8	Deduce the efficiency of an algorithm in the context of its use.	3	Students should understand and explain the difference in efficiency between a single loop, nested loops, a loop that ends when a condition is met or questions of similar complexity. Students should also be able to suggest changes in an algorithm that would improve efficiency, for example, using a flag to stop a search immediately when an item is found, rather than continuing the search through the entire list.
4.2.9	Determine the number of times a step in an algorithm will be performed for given input data.	3	Examination questions will involve specific algorithms (in pseudocode/ flow charts), and students may be expected to give an actual number (or range of numbers) of iterations that a step will execute.

4.3 introduction to programming (13 hours)

	Assessment statement	Obj	Teacher's notes
Nature of programming languages			
4.3.1	State the fundamental operations of a computer.	1	These include: add, compare, retrieve and store data. Complex capabilities are composed of very large numbers of very simple operations.
4.3.2	Distinguish between fundamental and compound operations of a computer.	2	For example, "find the largest" is a compound operation.

	Assessment statement	Obj	Teacher's notes
4.3.3	Explain the essential features of a computer language.	3	For example, fixed vocabulary, unambiguous meaning, consistent grammar and syntax. **TOK** Language and meaning.
4.3.4	Explain the need for higher level languages.	3	For example, as the human needs for computer systems have expanded it is necessary to abstract from the basic operations of the computer. It would take far too long to write the type of systems needed today in machine code.
4.3.5	Outline the need for a translation process from a higher level language to machine executable code.	2	For example, compiler, interpreter, virtual machine.
Use of programming languages Sub-programmes and objects support abstraction, which facilitates: ease of debugging and maintenance, reuse of code, modularity. There is no programming language specified in the SL/HL core. However, students must use a language that supports the basic constructs on the approved notation sheet.			
4.3.6	Define the terms: variable, constant, operator, object.	1	
4.3.7	Define the operators =, ≠, <, <=, >, >=, mod, div.	1	**LINK** Approved notation sheet.
4.3.8	Analyse the use of variables, constants and operators in algorithms.	3	For example, identify and justify the use of a constant as opposed to a variable in a given situation. **MYP** Mathematics: forms of numbers, algebra—patterns and sequences, logic, algorithms.
4.3.9	Construct algorithms using loops, branching.	3	Teachers must ensure algorithms use the symbols from the approved notation sheet. **LINK** Approved notation sheet. **MYP** Mathematics: using flow charts to solve problems in real-life contexts, logic, algorithms **MYP** Technology: design cycle (inputs, processes, outputs, feedback, iteration). **LINK** Connecting computational thinking and program design.

	Assessment statement	Obj	Teacher's notes
4.3.10	Describe the characteristics and applications of a collection.	2	Characteristics: • Contains similar elements. **LINK** HL extension, recursive thinking. **LINK** General principles of computational thinking, connecting computational thinking and program design.
4.3.11	Construct algorithms using the access methods of a collection.	3	**LINK** Connecting computational thinking and program design.
4.3.12	Discuss the need for sub-programmes and collections within programmed solutions.	3	Show an understanding of the usefulness of reusable code and program organization for the individual programmer, team members and future maintenance. **LINK** General principles of computational thinking, connecting computational thinking and program design. **MYP** Technology: use of software such as Alice.
4.3.13	Construct algorithms using pre-defined sub-programmes, one-dimensional arrays and/or collections.	3	**MYP** Mathematics: using flow charts to solve problems in real-life contexts, logic, algorithms. **MYP** Technology: design cycle (inputs, processes, outputs, feedback, iteration); use of software such as Alice. Students will only be required to analyse flow charts in the externally assessed components. Students will be expected to write and analyse pseudocode in the externally assessed components. **S/E**, **AIM 8** Appreciate the implications of using available code from sources such as online forums. **LINK** Connecting computational thinking and program design.

HL extension (45 hours)

Topic 5—Abstract data structures (23 hours)

5.1 Abstract data structures (23 hours)

This will be examined at the level of diagrams and pseudocode.

Students should be able to describe the most common data structures (arrays, stacks, queues, linked lists, binary trees) and the most common data processing operations on each of the basic data structures (addition, deletion and retrieval of data, traversal, searching for a given data, sorting of data into some order).

This should be taught and connected to flow charts, pseudocode and programming in the SL/HL core.

The basic ideas and their application should be illustrated with non-computer examples. Each basic idea should then be practised in specific algorithmic contexts using concepts drawn from flow charts, pseudocode and programming.

	Assessment statement	Obj	Teacher's notes
Thinking recursively			
5.1.1	Identify a situation that requires the use of recursive thinking.	2	Suggested practical activity: snowflakes and fractals, towers of Hanoi.
5.1.2	Identify recursive thinking in a specified problem solution.	2	**LINK** Binary trees.
5.1.3	Trace a recursive algorithm to express a solution to a problem.	2	Students will be required to state the output of the recursive algorithm. For example, trees. **LINK** Binary trees.
Abstract data structures			
5.1.4	Describe the characteristics of a two-dimensional array.	2	**LINK** One-dimensional arrays and basic algorithms.
5.1.5	Construct algorithms using two-dimensional arrays.	3	**LINK** Pseudocode information.
5.1.6	Describe the characteristics and applications of a stack.	2	Characteristics: • Last in, first out (LIFO). Examples of the applications of stacks may include running recursive processes, return memory addresses. **LINK** Recursive thinking; connecting computational thinking and program design.

	Assessment statement	Obj	Teacher's notes
5.1.7	Construct algorithms using the access methods of a stack.	3	Access methods: • push • pop • isEmpty. **LINK** Connecting computational thinking and program design.
5.1.8	Describe the characteristics and applications of a queue.	2	Characteristics: • First in, first out (FIFO). Examples of the applications of queues may include print queues and the computer modelling of physical queues (eg supermarket checkouts). Both linear and circular implementation of a queue are required. **LINK** Connecting computational thinking and program design.
5.1.9	Construct algorithms using the access methods of a queue.	3	Access methods: • enqueue • dequeue • isEmpty. **LINK** Connecting computational thinking and program design.
5.1.10	Explain the use of arrays as static stacks and queues.	3	Students should be able to explain push and pop operations, and test on empty/full stack. Students should be able to explain enqueue and dequeue operations, and test on empty/full queue.
Linked lists Linked lists will be examined at the level of diagrams and descriptions. Students are not expected to construct linked list algorithms using pseudocode.			
5.1.11	Describe the features and characteristics of a dynamic data structure.	2	Students should understand the concepts of nodes and pointer.
5.1.12	Describe how linked lists operate logically.	2	**LINK** Logical thinking.

	Assessment statement	Obj	Teacher's notes
5.1.13	Sketch linked lists (single, double and circular).	3	Students should be able to sketch diagrams illustrating: adding a data item to linked list, deleting specified data item, modifying the data held in the linked list, searching for a given data item.
Trees Binary trees will be examined at the level of diagrams and descriptions. Students are not expected to construct tree algorithms using pseudocode. Tracing and constructing algorithms are not expected.			
5.1.14	Describe how trees operate logically (both binary and non-binary).	2	**LINK** Recursive thinking.
5.1.15	Define the terms: parent, left-child, right-child, subtree, root and leaf.	1	These definitions only apply to a binary tree.
5.1.16	State the result of inorder, postorder and preorder tree traversal.	1	
5.1.17	Sketch binary trees.	3	Students should be able to sketch diagrams showing the resulting binary tree after adding a new data item, adding one or more new nodes, and/or removing one or more nodes.
Applications			
5.1.18	Define the term dynamic data structure.	1	
5.1.19	Compare the use of static and dynamic data structures.	3	**LINK** One-dimensional arrays.
5.1.20	Suggest a suitable structure for a given situation.	3	

Topic 6—Resource management (8 hours)

6.1 Resource management (8 hours)

	Assessment statement	Obj	Teacher's notes
System resources			
6.1.1	Identify the resources that need to be managed within a computer system.	2	Resources include: primary memory, secondary storage, processor speed, bandwidth, screen resolution, disk storage, sound processor, graphics processor, cache, network connectivity.

	Assessment statement	Obj	Teacher's notes
6.1.2	Evaluate the resources available in a variety of computer systems.	3	These should include: mainframes, servers, PCs, sub-laptops, as well as personal digital devices such as cell phones, PDAs and digital cameras. **AIM 9** Develop an appreciation of the issues linked to resource availability with continued developments in computer systems.
6.1.3	Identify the limitations of a range of resources in a specified computer system.	2	For example, single processor computers may not be able to render 3D graphics effectively.
6.1.4	Describe the possible problems resulting from the limitations in the resources in a computer system.	2	For example, user time wasted if the primary memory is too small or processor speed inadequate. Multi-access and multi-programming environments should be considered as well as single-user systems.
Role of the operating system			
6.1.5	Explain the role of the operating system in terms of managing memory, peripherals and hardware interfaces.	3	For example, allocating storage and keeping track of programs in memory, swapping between programs on time-slicing, priority or when one is waiting for input.
6.1.7	Outline OS resource management techniques: scheduling, policies, multitasking, virtual memory, paging, interrupt, polling.	2	Technical details as to how these are carried out will not be required, but it is expected that students will be familiar with these techniques and understand when and why they are used.
6.1.8	Discuss the advantages of producing a dedicated operating system for a device.	3	Advantages related to size, speed and customization should be considered. For example, using a dedicated operating system for a cell phone rather than using a pre-existing operating system. **S/E** Issue of proprietary software.

	Assessment statement	Obj	Teacher's notes
6.1.9	Outline how an operating system hides the complexity of the hardware from users and applications.	2	Students should be aware of a range of examples where operating systems virtualize real devices, such as drive letters, virtual memory, input devices, the Java virtual machine. **INT** Issue of localization causing compatibility problems between systems in different countries.

Topic 7—Control (14 hours)

7.1 Control (14 hours)

	Assessment statement	Obj	Teacher's notes
Centralized control systems			
7.1.1	Discuss a range of control systems.	3	A variety of control systems should be examined such as automatic doors, heating systems, taxi meters, elevators, washing machines, process control, device drivers, domestic robots, GPS systems, traffic lights and other common devices. Technical knowledge of specific systems is not expected but students should be able to analyse a specified system. **AIM 9** Develop an appreciation of the possibilities for control systems with developments in computer systems.
7.1.2	Outline the uses of microprocessors and sensor input in control systems.	2	These should be related to the examples suggested above.
7.1.3	Evaluate different input devices for the collection of data in specified situations.	3	Scenarios will be based on familiar situations to students.
7.1.4	Explain the relationship between a sensor, the processor and an output transducer.	3	Technical hardware details are not expected.
7.1.5	Describe the role of feedback in a control system.	2	**LINK** Connecting computational thinking and program design.

	Assessment statement	Obj	Teacher's notes
7.1.6	Discuss the social impacts and ethical considerations associated with the use of embedded systems.	3	**S/E** For example, tagging prisoners, surveillance, CCTV, improved safety systems.
Distributed systems			
7.1.7	Compare a centrally controlled system with a distributed system.	3	Technical hardware details are not expected.
7.1.8	Outline the role of autonomous agents acting within a larger system.	2	Technical hardware details are not expected.

Options (SL 30 hours/HL 45 hours)

Students must take one option.

A—Databases

A.1 Basic concepts (5 hours)

	Assessment statement	Obj	Teacher's notes
A.1.1	Outline the differences between data and information.	2	Data is meaningless. To be useful, data must be interpreted to produce information.
A.1.2	Outline the differences between an information system and a database.	2	Students must be aware that these terms are not synonymous. Databases are a component within an information system. **MYP** Technology.
A.1.3	Discuss the need for databases.	3	This should address topics such as the benefits of data sharing. **S/E** For example, correct information relating to customers and/or clients.
A.1.4	Describe the use of transactions, states and updates to maintain data consistency (and integrity).	2	For example, to ensure data consistency when moving money between two accounts it is necessary to complete two operations (debiting one account and crediting the other). Unless both operations are carried out successfully, the transaction will be rolled back. **S/E** For example, ensuring correct information relating to customers and/or clients.

	Assessment statement	Obj	Teacher's notes
A.1.5	Define the term database transaction.	1	
A.1.6	Explain concurrency in a data sharing situation.	3	
A.1.7	Explain the importance of the ACID properties of a database transaction.	3	ACID refers to: • atomicity • consistency • isolation • durability.
A.1.8	Describe the two functions databases require to be performed on them.	2	Query functions and update functions.
A.1.9	Explain the role of data validation and data verification.	3	

A.2 The relational database model (15 hours)

	Assessment statement	Obj	Teacher's notes
A.2.1	Define the terms: database management system (DBMS) and relational database management system (RDBMS).	1	
A.2.2	Outline the functions and tools of a DBMS.	2	A range of management functions and tools should be appreciated focusing on the creation, manipulation and interrogation of a database.
A.2.3	Describe how a DBMS can be used to promote data security.	2	Features involving data validation, access rights and data locking.
A.2.4	Define the term schema.	1	
A.2.5	Identify the characteristics of the three levels of the schema: conceptual, logical, physical.	2	
A.2.6	Outline the nature of the data dictionary.	2	
A.2.7	Explain the importance of a data definition language in implementing a data model.	3	
A.2.8	Explain the importance of data modelling in the design of a database.	3	

	Assessment statement	Obj	Teacher's notes
A.2.9	Define the following database terms: table, record, field, primary key, secondary key, foreign key, candidate key, composite primary key, join.	1	These are the accepted terms. Table is equivalent to relation/file. Record is equivalent to tuple/row. Field is equivalent to attribute/column. Only knowledge of an inner join is required.
A.2.10	Identify the different types of relationships within databases: one-to-one, one-to-many, many-to-many.	2	**AIM 4** Demonstrate initiative in applying thinking skills critically to understand the relationship between entities in a specified situation. **LINK** Thinking abstractly.
A.2.11	Outline the issues caused by redundant data.	2	**S/E**, **AIM 8** Issues relating to the integrity and reliability of data.
A.2.12	Outline the importance of referential integrity in a normalized database.	2	**S/E**, **AIM 8** Issues relating to the integrity and reliability of data.
A 2.13	Describe the differences between 1st Normal Form (1NF), 2nd Normal Form (2NF) and 3rd Normal Form (3NF).	2	For example: • 1NF has no repeating rows or columns. • 2NF is based on full functional dependency. • 3NF involves the removal of transitive dependencies.
A.2.14	Describe the characteristics of a normalized database.	2	Students will need to understand the characteristics of a database normalized to 3NF.
A.2.15	Evaluate the appropriateness of the different data types.	3	Students will be expected to be able to justify the selection of a particular data type in a specified situation. For example, integer or floating point. **S/E**, **AIM 8** The question of privacy for stakeholders. **S/E**, **AIM 8** The end-user must be seen as a key stakeholder when planning a new system. Comparing the different needs of each stakeholder. Who is a relevant stakeholder?

	Assessment statement	Obj	Teacher's notes
A.2.16	Construct an entity-relationship diagram (ERD) for a given scenario.	3	Students will be expected to construct entity-relationship diagrams in 3NF for a relational database. **AIM 4** Demonstrate skills enabling an understanding of the relationship between entities in a specified situation. **LINK** Thinking abstractly. **MYP** Technology: databases.
A.2.17	Construct a relational database to 3NF using objects such as tables, queries, forms, reports and macros.	3	Students will be expected to demonstrate knowledge of database designs in the SL/HL paper 2 that have resulted from practical activities. **TOK** Utilitarianism, the greatest good for the greatest number. The ends justify the means. **AIM 4**, **AIM 6** Demonstrate initiative in applying thinking and problem-solving skills critically to understand the relationship between entities in a specified situation. **AIM 5** The need to collaborate effectively with the end-user to resolve complex problems. **S/E**, **AIM 8** An awareness of the social impacts and ethical considerations when developing systems that potentially provide access to sensitive data.
A.2.18	Explain how a query can provide a view of a database.	3	
A.2.19	Describe the difference between a simple and complex query.	2	Students will be expected to be able to: • use Boolean operators such as AND, OR, NOT • create parameter queries • create derived fields. **MYP** Mathematics: forms of numbers, algebra—patterns and sequences, logic, algorithms.

	Assessment statement	Obj	Teacher's notes
A.2.20	Outline the different methods that can be used to construct a query.	2	Students will not be expected to be able to write queries in SQL. Students are expected to be aware of the language as a tool for data interrogation. **MYP** Mathematics: forms of numbers, algebra—patterns and sequences, logic, algorithms.

A.3 Further aspects of database management (10 hours)

	Assessment statement	Obj	Teacher's notes
A.3.1	Explain the role of a database administrator.	3	A visiting speaker or the school network manager could discuss his or her role. **S/E** Issues relating to privacy, security and integrity of data. **LINK** Systems in organizations.
A.3.2	Explain how end-users can interact with a database.	3	Practical activities to demonstrate SQL, QBE, visual queries, natural language interfaces.
A.3.3	Describe different methods of database recovery.	2	**S/E** Issues relating to the cost of implementing such systems weighed against the importance of the data.
A.3.4	Outline how integrated database systems function.	3	**AIM 9** An appreciation of the continued developments in computer systems.
A.3.5	Outline the use of databases in areas such as stock control, police records, health records, employee data.	2	**S/E** Issues relating to privacy, security and integrity of data. **LINK** Systems in organizations.
A.3.6	Suggest methods to ensure the privacy of the personal data and the responsibility of those holding personal data not to sell or divulge it in any way.	3	Students must be aware of the implications of large database systems. Principles of legislation such as Data Protection Act and Computer Misuse Act should be addressed (it is not necessary to study country-specific legislation). **S/E** Issues relating to privacy, security and integrity of data. **LINK** Systems in organizations.

	Assessment statement	Obj	Teacher's notes
A.3.7	Discuss the need for some databases to be open to interrogation by other parties (police, government, etc).	3	**S/E** Issues relating to privacy, security and integrity of data. **LINK** Systems in organizations. **AIM 8** An awareness of the social impacts and ethical considerations of holding large quantities of data.
A.3.8	Explain the difference between data matching and data mining.	3	

HL Extension

A.4 Further database models and database analysis (15 hours)

	Assessment statement	Obj	Teacher's notes
A.4.1	Describe the characteristics of different database models.	3	Database models should include: • relational • object-oriented • network • spatial • multi-dimensional. Students will be expected to refer to actual examples in their descriptions.
A.4.2	Evaluate the use of object-oriented databases as opposed to relational databases.	3	This may include references to data definition, manipulation and integrity.
A.4.3	Define the term data warehouse.	1	Subject oriented, integrated, time-variant and non-volatile collection of data used in decision-making.
A.4.4	Describe a range of situations suitable for data warehousing.	2	For example, strategic planning, business modelling.
A.4.5	Explain why data warehousing is time dependent.	3	Data in a warehouse is only valid for a period of time.
A 4.6	Describe how data in a warehouse is updated in real time.	2	Data is refreshed from data in operational systems.
A.4.7	Describe the advantages of using data warehousing.	2	A single manageable structure to support decision-making. Allows complex queries to be run across a number of business areas.

	Assessment statement	Obj	Teacher's notes
A.4.8	Explain the need for ETL processes in data warehousing.	3	Students should understand that processes are necessary to **E**xtract data from disparate sources, **T**ransform the data into a uniform format for specialized processing and **L**oad the extracted data into the data warehouse.
A.4.9	Describe how ETL processes can be used to clean up data for a data warehouse.	2	Examples should be used to show how disparate data can be changed to a uniform format in order to be suitable for analysis.
A.4.10	Compare the different forms of discovering patterns using data mining.	3	Students are expected to be able to describe the conceptual approach used by: • cluster analysis • associations • classifications • sequential patterns • forecasting. The student does not need to understand the detailed implementation of these methods. **AIM 8** An awareness of the social impacts and ethical considerations when data mining.
A.4.11	Describe situations that benefit from data mining.	2	Examples can be cited such as the use of mining techniques by banks to identify fraudulent credit card use; retailers can use mining techniques to identify subsets of the population likely to respond to a particular promotion.
A.4.12	Describe how predictive modelling is used.	2	The use of classification techniques such as "decision tree induction" or "backpropogation in neural networks". The determination of values for rows of a database useful for predictions.
A 4.13	Explain the nature of database segmentation.	3	The partitioning of a database according to some feature in common in the rows.
A 4.14	Explain the nature and purpose of link analysis.	3	The use of rules to establish associations between individual records in a data set.

	Assessment statement	Obj	Teacher's notes
A 4.15	Describe the process of deviation detection.	2	The detection of outlying data can be subjected to statistical techniques in order to identify unusual events or data subsets.

B—Modelling and simulation

B.1 The basic model (8 hours)

Students are expected to use a range of standard spreadsheet software in a variety of ways to create models. There is no need to purchase additional specialist software.

	Assessment statement	Obj	Teacher's notes
B.1.1	Define the term computer modelling.	1	
B.1.2	Identify a system that can be modelled.	2	Simple examples would involve financial planning, population growth, climate change, building design, engineering design, etc. Other situations could be modelling a game such as checkers or Mancala.
B.1.3	Identify the variables required to model a given system.	2	In an examination students will not be expected to identify more than four variables in a given system. **AIM 4** Applying thinking skills to identify variables and resolve a specified problem.
B.1.4	Describe the limitations of computer (mathematical) models.	2	In many situations it is not possible to know all of the variables involved. **AIM 9** An appreciation of the limitations of computer (mathematical) models. **MYP** Mathematics: forms of numbers, algebra—patterns and sequences, logic, algorithms.
B.1.5	Outline sensible grouping for collections of data items, including sample data.	2	For example, if age, height and weight are recorded for each person, group these as individual cells in a row in a table, or as items in parallel lists. **MYP** Mathematics: forms of numbers, algebra—patterns and sequences, logic, algorithms.
B.1.6	Design test-cases to evaluate a model.	3	**MYP** Design cycle.

	Assessment statement	Obj	Teacher's notes
B.1.7	Discuss the effectiveness of a test-case in a specified situation.	3	
B.1.8	Discuss the correctness of a model by comparing generated results with data that were observed in the original problem.	3	**AIM 6** Develop logical and critical thinking to discuss the correctness of a model.

B.2 Simulations (14 hours)

	Assessment statement	Obj	Teacher's notes
B.2.1	Define the term simulation.	1	
B.2.2	Explain the difference between a model and a simulation.	3	
B.2.3	Describe rules that process data appropriately and that produce results.	2	Rules may be presented as mathematical formulae, pseudocode algorithms, tables of input and output values (conversions), or in any other clear, understandable and specific format (eg a detailed English description). **MYP** Mathematics: forms of numbers, algebra—patterns and sequences, logic, algorithms.
B.2.4	Discuss rules and data representations and organization.	3	Students are expected to make critical comments, for example whether data has been sensibly organized and whether rules have been described correctly. **MYP** Mathematics: forms of numbers, algebra—patterns and sequences, logic, algorithms.
B.2.5	Construct simple models that use different forms of data representation and organization.	3	Students are expected to be familiar with spreadsheets in the SL/HL paper 2. Students are expected to have experience of other modelling software. **MYP** Mathematics: forms of numbers, algebra—patterns and sequences, logic, algorithms.

	Assessment statement	Obj	Teacher's notes
B.2.6	Design test-cases to evaluate a simulation program.	3	**MYP** Design cycle. **LINK** Connecting computational thinking and program.
B.2.7	Outline the software and hardware required for a simulation.	2	**MYP** Technology.
B.2.8	Describe changes in rules, formulae and algorithms that would improve the correspondence between results and observed data.	2	**MYP** Mathematics: forms of numbers, algebra—patterns and sequences, logic, algorithms. **MYP** Design cycle. **LINK** Connecting computational thinking and program.
B.2.9	Construct examples of simulations that involve changes in rules, formulae and algorithms.	3	**LINK** Connecting computational thinking and program.
B.2.10	Describe changes in data collection that could improve the model or simulation.	2	These changes could include ideas such as collecting different more data, or organizing and representing data differently. **MYP** Design cycle.
B.2.11	Discuss the reliability of a simulation by comparing generated results with data that were observed in the original problem.	3	**MYP** Design cycle.
B.2.12	Outline the advantages and disadvantages of simulation in a given situation rather than simply observing a real-life situation.	2	
B.2.13	Discuss advantages and disadvantages of using a simulation for making predictions.	3	Discuss the social consequences and ethical issues of the use of simulations. **AIM 8** An awareness of the social impacts and ethical considerations of using computer systems. **AIM 9** An appreciation of the increased accuracy of simulations as computer systems develop.

B.3 Visualization (8 hours)

	Assessment statement	Obj	Teacher's notes
B.3.1	Define the term visualization.	1	
B.3.2	Identify a two-dimensional use of visualization.	2	
B.3.3	Outline the memory needs of 2D visualization	2	
B.3.4	Identify a three-dimensional use of visualization.	2	
B.3.5	Outline the relationship between the images in memory and the 3D visualization.	2	Students do not need to know all the mathematical details but should understand the concept of rendering and be familiar with the terms wire-framing, ray tracing, lighting, key frame, mapping, texture.
B.3.6	Discuss the time and memory considerations of 3D animation in a given scenario.	3	In an examination the situation would be clearly described. **AIM 9** An appreciation of the implications on system resources as 3D animation packages become more complex.

HL Extension

B.4 Communication modelling and simulation (15 hours)

	Assessment statement	Obj	Teacher's notes
B.4.1	Outline the use of genetic algorithms.	2	Genetic algorithms develop the solution to a problem by starting with a random set of solutions and repeatedly using a subset of these to generate a better one. Students need to be familiar with the term fitness function. Examples are those where the suitability of the solutions can be measured, for example the travelling salesman problem.

	Assessment statement	Obj	Teacher's notes
B.4.2	Outline the structure of neural networks.	2	Students need to understand that neural networks are based on the knowledge of biological networks, but they will not be examined on the relationship. A simple block diagram showing inputs, hidden units and outputs and a brief is sufficient detail. **LINK** Connecting computational thinking and program.
B.4.3	Compare applications that use neural network modelling.	3	Examples include speech recognition, optical character recognition, natural language processing.
B.4.4	Compare different ways in which neural networks can be used to recognize patterns.	3	The concepts of supervised and unsupervised learning should be applied to the examples above and to any clearly explained example presented in the examination, as should the importance of genetic algorithms. **MYP** Mathematics: forms of numbers, algebra—patterns and sequences, logic, algorithms. **LINK** Connecting computational thinking and program.
B.4.5	Identify the key structures of natural language.	2	The terms noun, verb, syntax and semantics should be understood.
B.4.6	Discuss the differences between human and machine learning when related to language.	3	Using knowledge, such as the syntax of a language, leads to an appreciation of the difficulties involved in machine language learning. Students should be familiar with the concept of cognitive learning and the use of heuristics and probabilities in machine learning. **TOK** How can a machine know how it learns?
B.4.7	Outline the evolution of modern machine translators.	2	

	Assessment statement	Obj	Teacher's notes
B.4.8	Describe the role of chatbots to simulate conversation.	2	Students should be encouraged to use and analyse such standards as Eliza, Alice and Jabberwacky (as well as any more recent acknowledged ones) to compare conversations. **AIM 9** An appreciation of the possibilities associated with continued developments in computer systems. **MYP** Technology: software such as Alice.
B.4.9	Discuss the latest advances in natural language processing.	3	**AIM 9** An appreciation of the possibilities associated with continued developments in computer systems.

C—Web science

C.1 Creating the web (8 hours)

Students will be expected to have completed practical activities linked to developing different types of web pages and be able to evaluate when a particular type of web page is most appropriate.

	Assessment statement	Obj	Teacher's notes
C.1.1	Distinguish between the internet and World Wide Web (web).	2	
C.1.2	Describe how the web is constantly evolving.	2	Students will be expected to be aware of the major differences between the early forms of the web, Web 2.0, the semantic web and later developments. **MYP** Technology: searching the internet. **AIM 9** Develop an appreciation of the possibilities and limitations associated with the evolution of the web.

	Assessment statement	Obj	Teacher's notes
C.1.3	Identify the characteristics of the following: • hypertext transfer protocol (HTTP) • hypertext transfer protocol secure (HTTPS) • hypertext mark-up language (HTML) • uniform resource locator (URL) • extensible mark-up language (XML) • extensible stylesheet language transformations (XSLT) • JavaScript. • cascading style sheet (CSS).	2	
C.1.4	Identify the characteristics of the following: • uniform resource identifier (URI) • URL.	2	
C.1.5	Describe the purpose of a URL.	2	
C.1.6	Describe how a domain name server functions.	2	
C.1.7	Identify the characteristics of: • internet protocol (IP) • transmission control protocol (TCP) • file transfer protocol (FTP).	2	
C.1.8	Outline the different components of a web page.	2	To include features such as meta-tags, title, etc.
C.1.9	Explain the importance of protocols and standards on the web.	3	**INT** Protocols enable compatibility through a common "language" internationally.
C.1.10	Describe the different types of web page.	2	This should include examples such as personal pages, blogs, search engine pages, forums.
C.1.11	Explain the differences between a static web page and a dynamic web page.	3	To include analysis of static HTML web pages and dynamic web pages, eg PHP, ASP.NET, Java Servlets.
C.1.12	Explain the functions of a browser.	3	

	Assessment statement	Obj	Teacher's notes
C.1.13	Evaluate the use of client-side scripting and server-side scripting in web pages.	3	
C.1.14	Describe how web pages can be connected to underlying data sources.	2	Students will not be expected to write code to indicate how the connection is made, but should understand the principles of connecting to an underlying data source.
C.1.15	Describe the function of the common gateway interface (CGI).	2	
C.1.16	Evaluate the structure of different types of web pages.	3	

C.2 Searching the web (6 hours)

	Assessment statement	Obj	Teacher's notes
C.2.1	Define the term search engine.	1	
C.2.2	Distinguish between the surface web and the deep web.	2	**TOK** Data is always accessible.
C.2.3	Outline the principles of searching algorithms used by search engines.	2	Students will be expected to understand only the principles of the PageRank and HITS algorithms. **LINK** General principles of computational thinking, connecting computational thinking and program design.
C.2.4	Describe how a web crawler functions.	2	Teachers should be aware of the range of terms that can be associated with web crawlers such as bots, web spiders, web robots.
C.2.5	Discuss the relationship between data in a meta-tag and how it is accessed by a web crawler.	3	Students should be aware that this is not always a transitive relationship. **TOK** Data may not always have the intended meaning.
C.2.6	Discuss the use of parallel web crawling.	3	
C.2.7	Outline the purpose of web-indexing in search engines.	2	

	Assessment statement	Obj	Teacher's notes
C.2.8	Suggest how web developers can create pages that appear more prominently in search engine results.	3	Students will be expected to test specific data in a range of search engines, for example examining time taken, number of hits, quality of returns.
C.2.9	Describe the different metrics used by search engines.	2	Students will be expected to test specific data in a range of search engines, for example examining time taken, number of hits, quality of returns. **LINK** General principles of computational thinking. Connecting computational thinking and program design. **S/E**, **AIM 8** An understanding of search engine metrics could lead to exploitation.
C.2.10	Explain why the effectiveness of a search engine is determined by the assumptions made when developing it.	3	Students will be expected to understand that the ability of the search engine to produce the required results is based primarily on the assumptions used when developing the algorithms that underpin it. **LINK** Connecting computational thinking and program design.
C.2.11	Discuss the use of white hat and black hat search engine optimization.	3	**S/E**, **AIM 8** Developers of search engines should have a moral responsibility to produce an objective page ranking.
C.2.12	Outline future challenges to search engines as the web continues to grow.	2	Issues such as error management, lack of quality assurance of information uploaded. **AIM 9** Develop an appreciation that search engines will need to evolve to remain effective as the web grows.

C.3 Distributed approaches to the web (6 hours)

	Assessment statement	Obj	Teacher's notes
C.3.1	Define the terms: mobile computing, ubiquitous computing, peer-2-peer network, grid computing.	1	
C.3.2	Compare the major features of: • mobile computing • ubiquitous computing • peer-2-peer network • grid computing.	3	**LINK** Networks.
C.3.3	Distinguish between interoperability and open standards.	2	
C.3.4	Describe the range of hardware used by distributed networks.	2	Students should be aware of developments in mobile technology that have facilitated the growth of distributed networks.
C.3.5	Explain why distributed systems may act as a catalyst to a greater decentralization of the web.	3	**INT** Decentralization has increased international-mindedness.
C.3.6	Distinguish between lossless and lossy compression.	2	Students will not be required to study the detailed compression algorithms.
C.3.7	Evaluate the use of decompression software in the transfer of information.	3	Students can test different compression methods to evaluate their effectiveness.

C.4 The evolving web (10 hours)

	Assessment statement	Obj	Teacher's notes
C.4.1	Discuss how the web has supported new methods of online interaction such as social networking.	3	Students should be aware of issues linked to the growth of new internet technologies such as Web 2.0 and how they have shaped interactions between different stakeholders of the web. **S/E**, **AIM 8** Emerging technologies are modifying users' behaviour.
C.4.2	Describe how cloud computing is different from a client-server architecture.	2	Student should address the major differences only. **LINK** Networks.

	Assessment statement	Obj	Teacher's notes
C.4.3	Discuss the effects of the use of cloud computing for specified organizations.	3	To include public and private clouds. **S/E**, **AIM 8** Cloud computing could potentially conflict with privacy.
C.4.4	Discuss the management of issues such as copyright and intellectual property on the web.	3	Students should investigate sites such as TurnItIn and Creative Commons.
C.4.5	Describe the interrelationship between privacy, identification and authentication.	2	
C.4.6	Describe the role of network architecture, protocols and standards in the future development of the web.	2	**LINK** Networks. **AIM 9** Develop an appreciation that the future development of the web will have an effect on the rules and structures that support it.
C.4.7	Explain why the web may be creating unregulated monopolies.	3	**INT**, **S/E**, **AIM 8** The web is creating new multinational online oligarchies.
C.4.8	Discuss the effects of a decentralized and democratic web.		**S/E**, **INT** The web has changed users' behaviours and "removed" international boundaries.

HL Extension (15 hours)

C.5 Analysing the web (5 hours)

	Assessment statement	Obj	Teacher's notes
C.5.1	Describe how the web can be represented as a directed graph.	2	The vertices (nodes) represent web pages and the edges represent hyperlinks. It is not a complete graph. The directed graph formed by the web is known as the web graph. **LINK** Mathematics: graph theory.
C.5.2	Outline the difference between the web graph and sub-graphs.	2	A sub-graph will be assumed to be a set of pages linked to one specific topic.
C.5.3	Describe the main features of the web graph such as bowtie structure, strongly connected core (SCC), diameter.	2	Students must be aware the web has a structure that has emerged from the behaviour of web users.

	Assessment statement	Obj	Teacher's notes
C.5.4	Explain the role of graph theory in determining the connectivity of the web.	3	**LINK** Mathematics: graph theory.
C.5.5	Explain that search engines and web crawling use the web graph to access information.	3	Students should be aware of the Page Rank algorithm and explain how it works. No calculations are required.
C.5.6	Discuss whether power laws are appropriate to predict the development of the web.	3	

C.6 The intelligent web (10 hours)

	Assessment statement	Obj	Teacher's notes
C.6.1	Define the term semantic web.	1	
C.6.2	Distinguish between the text-web and the multimedia-web.	2	The traditional web is seen as being text based, the semantic web is multimedia based. **AIM 9** Develop an appreciation of the possibilities and limitations associated with the continuing evolution of the web.
C.6.3	Describe the aims of the semantic web.	2	**S/E**, **AIM 8** Emerging technologies are modifying users' behaviour.
C.6.4	Distinguish between an ontology and folksonomy.	2	
C.6.5	Describe how folksonomies and emergent social structures are changing the web.	2	**S/E**, **AIM 8** Emerging technologies are modifying users' behaviour.
C.6.6	Explain why there needs to be a balance between expressivity and usability on the semantic web.	3	**S/E**, **AIM 8** Emerging technologies are modifying users' behaviour.
C.6.7	Evaluate methods of searching for information on the web.	3	Teachers must address issues relating to searching for non-text based files/multimedia files such as using feature analysis.
C.6.8	Distinguish between ambient intelligence and collective intelligence.	2	

	Assessment statement	Obj	Teacher's notes
C.6.9	Discuss how ambient intelligence can be used to support people.	3	Students will be expected to have researched examples such as biometrics, nanotechnologies. **AIM 9** Develop an appreciation of the possibilities that ambient intelligence provides in supporting people when carrying out routine tasks.
C.6.10	Explain how collective intelligence can be applied to complex issues.	3	Students will be expected to have researched examples such as climate change, social bookmarking and stock market fluctuations. **AIM 5** Engender an awareness that effective collaboration and communication can resolve complex problems. **S/E**, **AIM 8** Emerging technologies are modifying users' behaviour. **TOK** It is possible to have a collective intelligence greater than the sum of the contributors.

D—Object-oriented programming

D.1 Objects as a programming concept (6 hours)

The paradigm of object-oriented programming should be introduced through discussion and example.

	Assessment statement	Obj	Teacher's notes
D.1.1	Outline the general nature of an object.	2	An object as an abstract entity and its components—data and actions. Familiar examples from different domains might be people, cars, fractions, dates and music tracks.
D.1.2	Distinguish between an object (definition, template or class) and instantiation.	2	Students must understand the difference in terms of code definitions, memory use and the potential creation of multiple instantiated objects.
D.1.3	Construct unified modelling language (UML) diagrams to represent object designs.	3	**LINK** Connecting computational thinking and program design.
D.1.4	Interpret UML diagrams.	3	**LINK** Connecting computational thinking and program design.

	Assessment statement	Obj	Teacher's notes
D.1.5	Describe the process of decomposition into several related objects.	2	A simple example with 3–5 objects is suggested. Examples related to D.1.1 could be employers, traffic simulation models, calculators, calendars, media collections. **LINK** Thinking abstractly. **AIM 4** Applying thinking skills critically to decompose scenarios.
D.1.6	Describe the relationships between objects for a given problem.	2	The relationships that should be known are dependency ("uses"), aggregation ("has a") and inheritance ("is a"). **LINK** Thinking abstractly. **AIM 4** Applying thinking skills critically to decompose scenarios.
D.1.7	Outline the need to reduce dependencies between objects in a given problem.	2	Students should understand that dependencies increase maintenance overheads.
D.1.8	Construct related objects for a given problem.	3	In examinations problems will require the students to construct definitions for no more than three objects and to explain their relationships to each other and to any additional classes defined by the examiners. **LINK** Connecting computational thinking and program design. **AIM 4** Applying thinking and algorithmic skills to resolve problems.
D.1.9	Explain the need for different data types to represent data items.	3	The data types will be restricted to integer, real, string and Boolean.
D.1.10	Describe how data items can be passed to and from actions as parameters.	2	Parameters will be restricted to pass-by-value of one of the four types in D.1.6. Actions may return at most one data item.

D.2 Features of OOP (4 hours)

Students should be able to describe the features of OOP that distinguish it from other approaches to computer programming.

	Assessment statement	Obj	Teacher's notes
D.2.1	Define the term encapsulation.	1	Data and actions are limited to the object in which they are defined.
D.2.2	Define the term inheritance.	1	A parent object holds common data and actions for a group of related child objects. Multiple inheritance is not required.
D.2.3	Define the term polymorphism.	1	Actions have the same name but different parameter lists and processes.
D.2.4	Explain the advantages of encapsulation.	3	For example, the scope of data should be confined to the object in which it is defined as far as possible in order to limit side effects and dependencies.
D.2.5	Explain the advantages of inheritance.	3	For example, a parent object holds common data and actions, which enhances reuse and reduces maintenance overheads.
D.2.6	Explain the advantages of polymorphism.	3	For example, an action in a child object may choose to override actions of a parent object. This allows an external program to use the same action on a family of objects without knowing the implementation detail.
D.2.7	Describe the advantages of libraries of objects.	2	For example, sorts and other complex algorithms and processes do not have to be "re-invented".
D.2.8	Describe the disadvantages of OOP.	2	For example, increased complexity for small problems; unsuited to particular classes of problem. **AIM 9** Develop an appreciation of the limitations of OOP.

	Assessment statement	Obj	Teacher's notes
D.2.9	Discuss the use of programming teams.	3	As compared to individuals working alone. Examples include speed to completion, information hiding to reduce module dependencies, expertise in narrow fields (eg testing, documentation), etc. **INT**, **AIM 5** The need to develop a common "language" to enable collaboration across international frontiers when resolving problems.
D.2.10	Explain the advantages of modularity in program development.	3	Advantages include easier debugging and testing, speedier completion, etc.

D.3 Program development (20 hours)

	Assessment statement	Obj	Teacher's notes
D.3.1	Define the terms: class, identifier, primitive, instance variable, parameter variable, local variable.	1	These are generally related to the object's data. See JETS.
D.3.2	Define the terms: method, accessor, mutator, constructor, signature, return value.	1	These are generally related to the object's actions. See JETS.
D.3.3	Define the terms: private, protected, public, extends, static.	1	These are generally related to the OOP features described in D.2. See JETS.
D.3.4	Describe the uses of the primitive data types and the reference class string.	2	In examination questions the primitive types will be limited to int, long, double, char and Boolean. **MYP** Mathematics: forms of numbers.
D.3.5	Construct code to implement assessment statements D.3.1–D.3.4.	3	Students may be asked to trace, explain or construct algorithms using the concepts associated with the terms.
D.3.6	Construct code examples related to selection statements.	3	Students may be asked to trace, explain or construct algorithms using simple and compound if … else constructs.
D.3.7	Construct code examples related to repetition statements.	3	Students may be asked to trace, explain or construct algorithms using for, while or do … while loops.
D.3.8	Construct code examples related to static arrays.	3	Students may be asked to trace, explain or construct algorithms using static arrays.

	Assessment statement	Obj	Teacher's notes
D.3.9	Discuss the features of modern programming languages that enable internationalization.	3	For example, use of UNICODE character sets. **INT** When organizations interact, particularly on an international basis, there may be issues of language differences.
D.3.10	Discuss the ethical and moral obligations of programmers.	3	For example, adequate testing of products to prevent the possibilities of commercial or other damage. Acknowledging the work of other programmers. The main aims of the Open Source movement should be known. **S/E AIM 8** An awareness of the ethical considerations when developing new code.

HL Extension

D.4 Advanced program development (15 hours)

	Assessment statement	Obj	Teacher's notes
D.4.1	Define the term recursion.	1	
D.4.2	Describe the application of recursive algorithms.	2	Students should understand that recursion can be applied to a small subset of programming problems to produce elegant solutions. Students should also understand that recursive algorithms are rarely used in practice. **LINK** Thinking abstractly, thinking recursively.
D.4.3	Construct algorithms that use recursion.	3	This is limited to a method that returns no more than one result and contains either one or two recursive calls. **LINK** Connecting computational thinking and program design.
D.4.4	Trace recursive algorithms.	2	All steps and calls must be shown clearly. **LINK** Connecting computational thinking and program design.
D.4.5	Define the term object reference.	1	As typified by simple classes that are self-referential.
D.4.6	Construct algorithms that use reference mechanisms.	3	

	Assessment statement	Obj	Teacher's notes
D.4.7	Identify the features of the abstract data type (ADT) list.	2	Students should understand the nature of an ADT—where no implementation details are known but the actions/methods are standard.
D.4.8	Describe applications of lists.	2	Students should understand that lists can be used to represent stacks and queues.
D.4.9	Construct algorithms using a static implementation of a list.	3	Lists will be restricted to singly linked types. Methods that should be known are add (head and tail), insert (in order), delete, list, isEmpty, isFull.
D.4.10	Construct list algorithms using object references.	3	Lists will be restricted to singly linked types. Methods that should be known are add (head and tail), insert (in order), delete, list, isEmpty, isFull.
D.4.11	Construct algorithms using the standard library collections included in JETS.	3	The classes are ArrayList and LinkedList. Students should have a broad understanding of the operation of these lists and their interface (methods) but not of the internal structure.
D.4.12	Trace algorithms using the implementations described in assessment statements D.4.9–D.4.11.	2	In examination questions, definitions of ArrayList and LinkedList methods will be given where necessary.
D.4.13	Explain the advantages of using library collections.	3	Students should understand that libraries provide convenient and reliable implementations of common programming tasks.
D.4.14	Outline the features of ADT's stack, queue and binary tree.	2	Students should be able to provide diagrams, applications and descriptions of these ADTs. For example, they should know that a binary tree can be used to efficiently store and retrieve unique keys.
D.4.15	Explain the importance of style and naming conventions in code.	3	Students should understand that meaningful identifiers, proper indentation and adequate comments all improve the readability of code for humans and thus save money, time and effort in programming teams. **INT**, **AIM 5** The need to develop a common "language" to enable collaboration across international frontiers when resolving problems.

Assessment in the Diploma Programme

General

Assessment is an integral part of teaching and learning. The most important aims of assessment in the Diploma Programme are that it should support curricular goals and encourage appropriate student learning. Both external and internal assessment are used in the Diploma Programme. IB examiners mark work produced for external assessment, while work produced for internal assessment is marked by teachers and externally moderated by the IB.

There are two types of assessment identified by the IB.

- Formative assessment informs both teaching and learning. It is concerned with providing accurate and helpful feedback to students and teachers on the kind of learning taking place and the nature of students' strengths and weaknesses in order to help develop students' understanding and capabilities. Formative assessment can also help to improve teaching quality, as it can provide information to monitor progress towards meeting the course aims and objectives.
- Summative assessment gives an overview of previous learning and is concerned with measuring student achievement.

The Diploma Programme primarily focuses on summative assessment designed to record student achievement at, or towards the end of, the course of study. However, many of the assessment instruments can also be used formatively during the course of teaching and learning, and teachers are encouraged to do this. A comprehensive assessment plan is viewed as being integral with teaching, learning and course organization. For further information, see the IB *Programme standards and practices* document.

The approach to assessment used by the IB is criterion-related, not norm-referenced. This approach to assessment judges students' work by their performance in relation to identified levels of attainment, and not in relation to the work of other students. For further information on assessment within the Diploma Programme please refer to the publication *Diploma Programme assessment: Principles and practice*.

To support teachers in the planning, delivery and assessment of the Diploma Programme courses a variety of resources can be found on the OCC or purchased from the IB store (http://store.ibo.org). Teacher support materials, subject reports, internal assessment guidance, grade descriptors, as well as resources from other teachers, can be found on the OCC. Specimen and past examination papers as well as markschemes can be purchased from the IB store.

Methods of assessment

The IB uses several methods to assess work produced by students.

Assessment criteria

Assessment criteria are used when the assessment task is open-ended. Each criterion concentrates on a particular skill that students are expected to demonstrate. An assessment objective describes what students should be able to do and assessment criteria describe how well they should be able to do it. Using assessment criteria allows discrimination between different answers and encourages a variety of responses.

Each criterion comprises a set of hierarchically ordered level descriptors. Each level descriptor is worth one or more marks. Each criterion is applied independently using a best-fit model. The maximum marks for each criterion may differ according to the criterion's importance. The marks awarded for each criterion are added together to give the total mark for the piece of work.

Markbands

Markbands are a comprehensive statement of expected performance against which responses are judged. They represent a single holistic criterion divided into level descriptors. Each level descriptor corresponds to a range of marks to differentiate student performance. A best-fit approach is used to ascertain which particular mark to use from the possible range for each level descriptor.

Markschemes

This generic term is used to describe analytic markschemes that are prepared for specific examination papers. Analytic markschemes are prepared for those examination questions that expect a particular kind of response and/or a given final answer from the students. They give detailed instructions to examiners on how to break down the total mark for each question for different parts of the response. A markscheme may include the content expected in the responses to questions or may be a series of marking notes giving guidance on how to apply criteria.

Assessment outline—SL

First examinations 2014

Assessment component	Weighting
External assessment (2 hours 30 minutes) **Paper 1 (1 hour 30 minutes)** Paper 1 is an examination paper consisting of **two compulsory sections**. • Section A (30 minutes approximately) consists of several **compulsory** short answer questions. The maximum mark for this section is 25. • Section B (60 minutes approximately) consists of three **compulsory** structured questions. The maximum mark for this section is 45. (70 marks)	**70%** **45%**
Paper 2 (1 hour) Paper 2 is an examination paper linked to the option studied. The paper consists of between two and five **compulsory** questions. (45 marks) **Calculators:** The use of calculators is **not** permitted in any computer science examination.	**25%**
Internal assessment (40 hours) This component is internally assessed by the teacher and externally moderated by the IB at the end of the course. **Solution (30 hours)** The development of a computational solution. Students must produce: • a cover page that follows the prescribed format • a product • supporting documentation (word limit 2,000 words). (34 marks) **Group 4 project (10 hours)** To be assessed using the criterion Personal skills. (6 marks) (total 40 marks)	**30%**

Assessment outline—HL

First examinations 2014

Assessment component	Weighting
External assessment (4 hours 30 minutes) **Paper 1 (2 hours 10 minutes)** Paper 1 is an examination paper consisting of **two compulsory sections**. • Section A (30 minutes approximately) consists of several **compulsory** short answer questions. The maximum mark for this section is 25. • Section B (100 minutes approximately) consists of five **compulsory** structured questions. The maximum mark for this section is 75. (100 marks)	**80%** **40%**
Paper 2 (1 hour 20 minutes) Paper 2 is an examination paper linked to the option studied. The paper consists of between three and seven **compulsory** questions. The SL/HL core questions are common and worth 45 marks, HL extension is worth 20 marks. (65 marks)	**20%**
Paper 3 (1 hour) Paper 3 is an examination paper of **1 hour** consisting of **four compulsory** questions based on a pre-seen case study. (30 marks) **Calculators:** The use of calculators is **not** permitted in any computer science examination.	**20%**

Assessment component	Weighting
Internal assessment (40 hours) This component is internally assessed by the teacher and externally moderated by the IB at the end of the course. **Solution (30 hours)** The development of a computational solution. Students must produce: • a cover page that follows the prescribed format • a product • supporting documentation (word limit 2,000 words). (34 marks) **Group 4 project (10 hours)** To be assessed using the criterion Personal skills. (6 marks) (total 40 marks)	**20%**

External assessment

Two different methods are used to assess students.

- Detailed markschemes specific to each examination paper
- Markbands

The markbands are published in this guide.

For paper 1, there are markschemes.

For paper 2, there are markschemes.

For paper 3, there are markbands and markschemes.

The markbands are related to the assessment objectives established for the computer science course and the group 4 grade descriptors. The markschemes are specific to each examination.

External assessment details—SL

Paper 1

Duration: 1 hour 30 minutes
Maximum mark: 70
Weighting: 45%

The purpose of the paper is to assess the student's ability to demonstrate the following objectives in relation to the syllabus:

- Assessment objective 1—know and understand
- Assessment objective 2—apply and use
- Assessment objective 3—construct, analyse and evaluate

Section A (30 minutes approximately) consists of several **compulsory** short answer questions testing mainly objectives 1 and 2. The maximum mark for this section is 25. Some of the questions are common to this paper and HL paper 1, section A.

Section B (60 minutes approximately) consists of three **compulsory** structured questions that may be subdivided. The maximum mark for this section is 45. Some questions may be common to this paper and HL paper 1, section B.

The number of marks for each part will be given on the paper, and is linked to the command term used. This will indicate to students the depth of the response required.

Paper 2

Duration: 1 hour
Maximum mark: 45
Weighting: 25%

The purpose of the paper is to assess the student's ability to demonstrate the following objectives in relation to the option chosen:

- Assessment objective 1—know and understand
- Assessment objective 2—apply and use
- Assessment objective 3—construct, analyse and evaluate

Students are required to answer all the questions for the option chosen. The questions are common to this paper and HL paper 2, section A.

Students are not expected to construct code in the following options:

- databases
- modelling and simulation
- web science.

Students will be expected to interpret and/or construct code in Java in the OOP option.

External assessment details—HL

Paper 1

Duration: 2 hours 10 minutes
Maximum mark: 100
Weighting: 40%

The purpose of the paper is to assess the student's ability to demonstrate the following objectives in relation to the syllabus:

- Assessment objective 1—know and understand
- Assessment objective 2—apply and use
- Assessment objective 3—construct, analyse and evaluate

Section A (30 minutes approximately) consists of several **compulsory** short answer questions testing mainly objectives 1 and 2. The maximum mark for this section is 25. Some of the questions are common to this paper and SL paper 1, section A.

Section B (100 minutes approximately) consists of five **compulsory** structured questions that may be subdivided. The maximum mark for this section is 75. Some questions may be common to this paper and SL paper 1, section B.

The number of marks for each part will be given on the paper, and is linked to the command term used. This will indicate to students the depth of the response required.

Paper 2

Duration: 1 hour 20 minutes
Maximum mark: 65
Weighting: 20%

The purpose of the paper is to assess the student's ability to demonstrate the following objectives in relation to the option chosen:

- Assessment objective 1—know and understand
- Assessment objective 2—apply and use
- Assessment objective 3—construct, analyse and evaluate

Section A consists of common questions (to the SL/HL core) for the option chosen. The maximum mark is 45.

Section B consists of questions for the option chosen relating to the HL extension. The maximum mark is 20.

Students are not expected to construct code in the following options:

- databases
- modelling and simulation
- web science.

Students will be expected to interpret and/or construct code in Java in the OOP option.

Paper 3

Duration: 1 hour
Maximum mark: 30
Weighting: 20%

The paper

This paper is based on a case study produced annually by the IB and available on the OCC.

A clean copy of the case study must be downloaded by the IB coordinator and issued to the students with their examination paper.

The purpose of the paper is to assess the student's ability to demonstrate the following objectives in relation to the syllabus:

- Assessment objective 1—know and understand
- Assessment objective 2—apply and use
- Assessment objective 3—construct, analyse, evaluate and formulate

This paper consists of four structured questions, which assess the whole syllabus in an integrated way.

Students are required to answer all of the questions.

The questions are related to the scenario in the case study. In addition to the case study further stimulus material may be provided in the examination paper.

Questions 1, 2 and 3 may be subdivided. Question 4 will require the synthesis of information from a range of sources, including independent research and investigations beyond the case study, to develop an extended response to a specified issue.

The number of marks for each part will be given on the paper, and is linked to the command term used. This will indicate to students the depth of the response required.

The case study

The case study is a valuable teaching tool that can be used to integrate all aspects of the syllabus.

This case study will be provided 12 months before the May examination session (18 months before the November session) so that students can carry out detailed research prior to the HL paper 3 examination, which contributes 20% of the final mark.

The computer science case study provides the stimulus material for the investigation of a scenario involving current developments and/or issues in computer science. The information obtained will prepare students and form the basis of the requirements for answering the questions in HL paper 3.

The case study is an opportunity to keep the course abreast of current technology by introducing new technical concepts or issues requiring a more in-depth investigation than that required in the rest of the course.

Through their investigation of the case study, students should be able to:

- demonstrate an understanding of the computer science concepts fundamental to the system(s) in the case study (objective 1)
- demonstrate an understanding of how the system(s) in the case study work (objective 1)
- apply material from the course syllabus in the context of the case study (objective 2)
- explain how scenarios specified in the case study may be related to other similar local and global scenarios (objective 3)
- discuss the social impacts and ethical issues relevant to the case study (objective 3)
- explain technical issues relating to the case study (objective 3)
- evaluate information that may be gathered from local and global sources including field trips, interviews, primary and secondary research, invited guest speakers and online interviews (objective 3)
- evaluate, formulate or justify strategic solutions based on the synthesis of information from the case study itself, additional research and new stimulus material provided in the examination paper (objective 3).

External assessment markbands—HL

All questions except for question 4 on HL paper 3 use indicative content with adaptable markschemes and markbands. The markband for question 4 on HL paper 3 appears below.

HL paper 3 question 4 markband

There must be evidence of independent research and investigation for students to reach the top level.

Marks	Level descriptor
No marks	• No knowledge or understanding of the relevant issues and concepts. • No use of appropriate terminology.
Basic 1–3 marks	• Minimal knowledge and understanding of the relevant issues or concepts. • Minimal use of appropriate terminology. • The answer may be little more than a list. • No reference is made to the information in the case study or independent research.

Marks	Level descriptor
Adequate 4–6 marks	• A descriptive response with limited knowledge and/or understanding of the relevant issues or concepts. • A limited use of appropriate terminology. • There is limited evidence of analysis. • There is evidence that limited research has been undertaken.
Competent 7–9 marks	• A response with knowledge and understanding of the relevant issues and/or concepts. • A response that uses terminology appropriately in places. • There is some evidence of analysis. • There is evidence that research has been undertaken.
Proficient 10–12 marks	• A response with a detailed knowledge and clear understanding of the relevant issues and/or concepts. • A response that uses terminology appropriately throughout. • There is competent and balanced analysis. • Conclusions are drawn that are linked to the analysis. • There is clear evidence that extensive research has been undertaken.

Internal assessment

Purpose of internal assessment

Internal assessment is an integral part of the course and is compulsory for both SL and HL students. It enables students to demonstrate the application of their skills and knowledge, and to pursue their personal interests, without the time limitations and other constraints that are associated with written examinations. The internal assessment should, as far as possible, be woven into normal classroom teaching and not be a separate activity conducted after a course has been taught.

The internal assessment requirements at SL and at HL are the same. However, these requirements contribute to a different percentage of the overall mark. Students are required to produce a solution that consists of a cover page, the product and the documentation. The focus of the solution is on providing either an original product or additional functionality to an existing product for a client.

The internal assessment component (solution), as well as being practical and productive, forms an important part of the assessment of the computer science course. It is imperative, therefore, that the teacher provides appropriate guidance to students.

Guidance and authenticity

The solution submitted for internal 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 internal assessment 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
- the ethical guidelines mentioned in the "Requirements and recommendations" section of this document
- the assessment criteria; students must understand that the work submitted for assessment must address these criteria effectively.

Teachers and students must discuss the internally assessed work. Students should be encouraged to initiate discussions with the teacher to obtain advice and information, and students must not be penalized for seeking guidance. However, if a student could not have completed the work without substantial support from the teacher, this should be recorded on the appropriate form from the *Handbook of procedures for the Diploma Programme*.

It is the responsibility of teachers to ensure that all students understand the basic meaning and significance of concepts that relate to academic honesty, 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 internally assessed work must be entirely their own.

As part of the learning process, teachers can give advice to students on a first draft of the internally assessed work. This advice should be in terms of the way the work could be improved, but this first draft must not be heavily annotated or edited by the teacher. The next version handed to the teacher after the first draft must be the final one.

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 sign the coversheet for internal assessment to confirm that the work is his or her authentic work and constitutes the final version of that work. Once a student has officially submitted the final version of the work to a teacher (or the coordinator) for internal assessment, together with the signed coversheet, it cannot be retracted.

Authenticity may be checked by discussion with the student on the content of the work, and scrutiny of one or more 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 requirement for teachers and students to sign the coversheet for internal assessment applies to the work of all students, not just the sample work that will be submitted to an examiner for the purpose of moderation. If the teacher and student sign a coversheet, but there is a comment to the effect that the work may not be authentic, the student will not be eligible for a mark in that component and no grade will be awarded. For further details refer to the IB publication *Academic honesty* and the relevant articles in the *General regulations: Diploma Programme*.

The same piece of work cannot be submitted to meet the requirements of both the internal assessment and the extended essay.

Group work

The development of the solution must be undertaken by the student on an individual basis. Collaborative or group work may not be undertaken by students.

Time allocation

It is recommended that a total of approximately 30 teaching hours for both SL and HL should be allocated to the work.

This should include:

- time for the teacher to explain to students the requirements of the internal assessment, including codes of ethical behaviour and confidentiality
- class time for students to work on the internal assessment
- time spent by the student making arrangements to collect data as appropriate
- time for consultation between the teacher and each student
- time to review and monitor progress, and to check authenticity.

Additional time may be needed outside normal class time for students to work independently, such as acquiring additional skills required for the project and consulting with other people.

Requirements and recommendations

Teachers and students will need to discuss issues relating to the design of the product, the collection of data and consultations with others. Students should be encouraged to initiate discussions with the teacher to obtain advice and information, and will not be penalized for seeking advice.

Ethical guidelines for internal assessment

Given the nature of the project, students must take into account ethical problems and implications for undertaking research and developing the solution, for example, ensuring the confidentiality and security of data. Wherever possible, original data should be used or be 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 solution before any investigation is begun.
- All data collected must be stored securely in order to maintain confidentiality.
- Only the data collected for the solution can be used. It must not be used for any other purpose without explicit permission.

Teachers should refer to the *Ethical practice in the Diploma Programme* poster for further guidance.

Health and safety guidelines

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

Word count

Students must produce a solution that includes supporting documentation up to a maximum of 2,000 words. If the word limit is exceeded, the teacher's assessment of the documentation must be based on the first 2,000 words.

Work that falls significantly beneath the stated word count is unlikely to fully meet the stated requirements of the task and is likely to receive low marks.

Using assessment criteria for internal assessment

For internal assessment, a number of assessment criteria have been identified. Each assessment criterion has level descriptors describing specific levels of achievement 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. Teachers should award the lower marks if the student's work demonstrates the qualities described to a lesser extent.
- Only whole numbers should be recorded; partial marks, that is 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 level of achievement 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

Solution

Duration: 30 hours
Weighting at SL: 30%
Weighting at HL: 20%

Introduction

The requirement of the internal assessment is to develop a solution for a specified client to a specified problem or an unanswered question.

The solution is assessed using five criteria.

- Planning
- Solution overview
- Development
- Functionality and extensibility of product
- Evaluation

Key terms

The term "solution" refers to all the work submitted by the student for the internal assessment; the term "product" refers to the completed software only. The product is a subset of the solution.

The terms "developer" and "student" are synonymous.

The term "client" refers to the person for whom the product is being developed. The student may also be the client.

The term "adviser" refers to a third party the student **must** identify to assist him or her in the development of the product.

There are three scenarios for the development of the product.

1. The student is developing the product for a third party who is the client and also acts as the adviser.
2. The student is developing the product for a third party who is the client. Another person acts as the adviser.
3. The student is the client (developing the product for himself or herself). An appropriate adult must act as the adviser.

Choice of topic

In identifying a problem, students can select any topic that interests them. It does not have to be directly related to the specified themes in the syllabus or to the option studied.

Students should undertake a challenging task using appropriate techniques to showcase their algorithmic thinking and organizational skills.

The solution may take one of these forms:

- Creating a new system, such as an OOP program, a relational database, a simulation or a stand-alone/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

It is essential that whatever form the solution takes it ensures the student can explicitly demonstrate and document his or her algorithmic thinking skills.

Examples are illustrated in the teacher support material.

It should be noted that products created using templates that show no evidence of modification in their structure, design or functionality are not permitted. Examples of **inappropriate** products include:

- the development of a programming product only using copied code
- the development of a website (product) using a web-based template that determines its structure and layout
- the use of unmodified exemplar products or templates provided with software such as the Northwind database in MS Access
- a product that does not meet the ethical requirements outlined in the "Requirements and recommendations" section of this document.

Choice of adviser

Students will need to work closely with the adviser throughout the development of the solution. Therefore it is recommended that wherever possible, students select an adviser who is known to them or their family. This could include members of the school community, local clubs or businesses.

Requirements

The internal assessment consists of three parts:

1. A cover page
   - The cover page form to be used is included in HTML format within the zip file available on the OCC.
   - The cover page must be submitted in HTML format and should provide access to the product and associated documentation via **relative** hyperlinks.
   - The cover page is not included in the overall word count for the project, nor is its functionality assessed.
   - Information to access or locate the product, for example, a username and password, must be provided in the cell on the cover page.
   - The cover page must be called [cand_no]_[cand_name]_CoverPage.htm and be located in the top level folder.
2. The product
3. The documentation including a video
   3.1 The video must be in a commonly used format such as .avi or .wmv.

All three of these must be submitted digitally for moderation.

Instructions for the submission of student work can be found in the *Handbook of procedures for the Diploma Programme*.

Components of the solution

Product

Students should aim to develop a product that uses appropriate (complex) techniques, is fully functional and allows the moderator, if possible, access to its complete internal structure.

The moderator must be able to see the product functioning as a video. The video should address the success criteria stated in the planning criterion.

Any text within the product is not included in the overall word count for the solution.

Documentation

This information must be read in conjunction with that in the "Organization of documentation" section.

A zip file is available on the OCC that contains the cover page and templates required for submitting the solution.

The final documentation consists of:

- **Information added to the *Record of tasks* form and in the information linked to the design overview**

  The information added must be in the following style(s):
  - bullet points or tables to list information
  - scanned diagrams or other appropriate images as part of the design process
  - other styles of non-extended writing or diagrammatic representation such as flow charts, Gantt charts or spider diagrams where appropriate.

If the student includes extended writing, the words will be included in the word count and the student will self-penalize if the total number of words in the documentation exceeds 2,000.

The *Record of tasks* form in the zip file must be used.

- **A series of documents** that use text (extended writing) that:
  - describes the scenario, client requirements and/or adviser input
  - justifies the rationale behind the proposed product
  - explains the techniques used to develop the product
  - evaluates the success of the product in resolving the original problem and client and/or adviser feedback
  - recommends further improvements to the product.

  This should be the **only** information that is included in the word count and it **must not exceed 2,000 words**.

  It is recommended that the blank files in the zip file are used for the documentation.
- **Appendix/appendices** that show, if appropriate, any additional information such as:
  - evidence of consultation with the client/adviser
  - evidence of feedback from the client/adviser
  - additional videos or documents containing screenshots of the product functioning.

Organization of documentation

The documentation must be located in the **Documentation folder**. It is associated with assessment criteria A—E and any additional material in the appendices of this guide.

The table below indicates the content and nature of each of the files and the criterion that it relates to.

Document	Method of submission	Criterion
Description of scenario	Extended writing	A Planning
Rationale for the proposed product	Extended writing	A Planning
Success criteria for product	Bullet points	A Planning
Record of tasks	*Record of tasks* form	B Solution overview
Design overview	Design overview document, for example, screenshots, flowcharts, tables, diagrams	B Solution overview
Developing the product	Extended writing with screenshot evidence	C Development
The functioning product	Video (2–7 minutes in length) demonstrating the product	D Functionality and extensibility of product
Extensibility of product	Assessed through design overview and developing the product	D Functionality and extensibility of product

Document	Method of submission	Criterion
Evaluating the product	Extended writing	E Evaluation
Recommendations for improving the product	Extended writing	E Evaluation
Appendix	To contain additional information if appropriate	n/a

Development of the solution

Students are advised to use the following guidelines to develop the solution. This will ensure it fulfills the requirements of the assessment criteria.

Prior to developing the product the teacher must have:

- approved the choice of client/adviser and scenario
- reviewed preliminary designs and checked that the:
  - templates provided in the zip file are being used
  - proposed timeline for completing the product is realistic
  - scope and nature of the product are appropriate.

Criterion A: Planning

The scenario

The following key questions should be considered.

- Who is the client/adviser?
- Is the choice of client/adviser appropriate?
- Why is the product being developed?

Rationale for proposed solution

The rationale behind the choice of the proposed product must be in **extended writing**, with reference to the student's consultations with the client and/or adviser, justifying how the choice of this particular product is an effective solution.

Before reaching a decision for the proposed product the student and/or computer science teacher must determine:

- if the student has the technical skills and access to the software required to develop the product
- whether the client's hardware and software is compatible with the product
- if the data required for the product can be obtained by either the student and/or appropriate third parties
- how any security implications for the development and operation of the product can be resolved.

Success criteria

The success criteria (that are evaluated in criterion E) should be listed in the form of bullet points.

If the student is the client, they must have an adviser who can review the success criteria and provide the validation of the product.

Criterion B: Solution overview

A *Record of tasks* form in the zip file must be used for the product proposed in criterion A.

The record of tasks form addresses:

- a chronology of the key events in planning, designing, developing, testing and implementing the solution
- any other issues that may arise that may affect the development of the solution.

The design overview should include:

- design methodologies appropriate to the type of product being designed
- different levels of draft design, such as the overall structure and the internal layout of the product itself; this can also include investigation into specific elements used within the product (such as classes, sub-classes, tables, queries, style sheets, graphic elements, effects)
- evidence of a testing plan that addresses the main areas of functionality of the product.

The student must submit final versions of the *Record of tasks* form and the design overview. However, the teacher will need to see earlier versions to determine whether the product proposed is appropriate and feasible.

Criterion C: Development

The product must be compatible with the information in criterion A and criterion B.

The student must present a list of the techniques used in developing the product.

The techniques may include algorithmic thinking, data structures, software tools and user interface. This list need not be exhaustive but should illustrate how the major components of the product were developed.

The student must provide evidence of algorithmic thinking.

The information in the development documentation must provide a detailed account, using extended writing and other appropriate information, to explain the following.

- The structure of the product and why it is appropriate
- The algorithmic thinking used in the development of the product
- The techniques used in the development of the product and reasons why they are appropriate to it (may include screenshots, exemplar data, reference to information in the appendix)
- Any existing tools that are used in the development of the product, such as code libraries, software packages, web hosting, security information or infrastructure issues

Any reference material such as templates, program code, applets or other materials that have been used or modified must be acknowledged in this criterion. The code used in the product can be included in the appendix.

Criterion D: Functionality and extensibility of product

This criterion should be completed as two parts and does not require any additional written documentation.

Functionality of the product

The student must use the video to demonstrate the product functioning. This evidence will be supported, where possible, by the product on the CD-ROM/DVD or USB.

Extensibility of product
The student should design the product so that it can be maintained by another party and/or be further developed. Therefore the design of the product should include appropriate folder and data structures, intuitive file and/or class naming conventions and, where appropriate, comments in the code.

Criterion E: Evaluation
This criterion should be completed as two parts.

Evaluation of the product
The evaluation of the product should refer directly to the success criteria in criterion A, feedback from the client/adviser, as well as any other appropriate feedback obtained.

Recommendations for the future development of the product
The student will use the feedback and the evaluation of the specific performance criteria to recommend possible future developments to the product. These recommendations should explain the benefits of these developments.

Internal assessment criteria—SL and HL

Rationale

General overview
The computer science internal assessment focuses on the balance between the level of algorithmic thinking and problem-solving required to develop a product within the framework of the design cycle.

The assessment criteria
Criteria A, B and E are process-oriented and examine how the internal assessment task was carried out and allow common assessment criteria to be applied to different types of product from the different options. Criterion C is a holistic assessment of the final product and assesses the student's understanding of the concepts involved in its development. Criterion D is a holistic assessment of the functionality and future extensibility of the product.

Criterion A: Planning (6 marks)
The success criteria identified in criterion A will be used in criterion D to evaluate the effectiveness of the product.

Marks	Description
0	The response does not reach a standard described by the descriptors below.
1–2	An appropriate scenario for investigation for an identified client is stated. The rationale for choosing the proposed product is identified. The criteria for evaluating the success of the product are generally inappropriate.

Marks	Description
3–4	An appropriate scenario for investigation for an identified client, providing evidence of consultation, is stated. The rationale for choosing the proposed product is partially explained and includes some appropriate criteria for evaluating the success of the product.
5–6	An appropriate scenario for investigation for an identified client, providing evidence of consultation, is described. The rationale for choosing the proposed product is justified and includes a range of appropriate criteria for evaluating the success of the product.

Criterion B: Solution overview (6 marks)

- The student must provide a record of tasks and a design overview that includes an outline test plan.
- The *Record of tasks* form must be used.
- The record of tasks and design overview must refer to the product proposed in criterion A.

Marks	Description
0	The response does not reach a standard described by the descriptors below.
1–2	The record of tasks and the design overview, including an outline test plan, are limited. From this information it is difficult to see how the product was developed.
3–4	The record of tasks and the design overview, including an outline test plan, are partially complete. They provide a basic understanding of how the product was developed.
5–6	The record of tasks and the design overview, including an outline test plan, are detailed and complete. From this information it is clear how the product was developed.

Criterion C: Development (12 marks)

- The student must identify techniques used in developing the product.
- The student must explain the techniques, with screenshots, that were used to develop the product identified in criterion A, explaining why they have been used and why they are adequate for the task.

Marks	Description
0	The response does not reach a standard described by the descriptors below.
1–4	The use of techniques demonstrates a low level of complexity and ingenuity or does not address the scenario identified in criterion A. It is characterized by limited use of existing tools. There is no explanation of why the techniques are used or how they are adequate for the task. Sources are used but are not identified.

Marks	Description
5–8	The use of techniques demonstrates a moderate level of complexity and ingenuity in addressing the scenario identified in criterion A. It is characterized by some appropriate use of existing tools. There is some attempt to explain the techniques used and why they are adequate for the task. All sources are identified.
9–12	The use of techniques demonstrates a high level of complexity and ingenuity in addressing the scenario identified in criterion A. It is characterized by the appropriate use of existing tools. The techniques are adequate for the task and their use is explained. All sources are identified.

Criterion D: Functionality and extensibility of product (4 marks)

This criterion assesses the extent to which the product:

- functions, as evidenced in the video
- can be expanded and modified by future users as evidenced in the design and development documentation.

Marks	Description
0	The response does not reach a standard described by the descriptors below.
1–2	The video shows that the product functions partially. Some expansion and modification of the product is possible but difficult.
3–4	The video shows that the product functions well. Some expansion and modifications of the product are straightforward.

Criterion E: Evaluation (6 marks)

- The student must evaluate the effectiveness of the product based on feedback from the client/adviser. This must include direct references to the success criteria identified in criterion A.
- The student must recommend proposals for the future improvement of the product.

Marks	Description
0	The response does not reach a standard described by the descriptors below.
1–2	There is a limited attempt to evaluate the product against the success criteria identified in criterion A. There is limited evidence of feedback from the client/adviser and any recommendations for further improvement are trivial or unrealistic.
3–4	The product is partially evaluated against the success criteria identified in criterion A including feedback from the client/adviser. Recommendations for further improvement of the product are largely realistic.
5–6	The product is fully evaluated against the success criteria identified in criterion A including feedback from the client/adviser. Recommendations for further improvement of the product are realistic.

Summary of the group 4 project

The group 4 project is a collaborative activity where students from different group 4 subjects work together on a scientific or technological topic, allowing for concepts and perceptions from across the disciplines to be shared. This is to encourage an understanding of the relationships between scientific disciplines and the overarching nature of the scientific method. The project can be practically or theoretically based. Collaboration between schools in different regions is encouraged.

The group 4 project allows students to appreciate the environmental, social and ethical implications of science and technology. It may also allow them to understand the limitations of scientific study, for example, the shortage of appropriate data and/or the lack of resources. The emphasis is on interdisciplinary cooperation and the processes involved in scientific investigation, rather than the products of such investigation.

The choice of scientific or technological topic is open but the project should clearly address the group 4 aims 7, 8 and 10 of the computer science subject guide.

Ideally, the project should involve students collaborating with those from other group 4 subjects at all stages. To this end, it is not necessary for the topic chosen to have clearly identifiable separate subject components. However, for logistical reasons some schools may prefer a separate subject "action" phase (see the following "Project stages" section).

Project stages

The 10 hours allocated to the group 4 project, which are part of the teaching time set aside for internal assessment, can be divided into three stages: planning, action and evaluation.

Planning

This stage is crucial to the whole exercise and should last about two hours.

- The planning stage could consist of a single session, or two or three shorter ones.
- This stage must involve all group 4 students meeting to "brainstorm" and discuss the central topic, sharing ideas and information.
- The topic can be chosen by the students themselves or selected by the teachers.
- Where large numbers of students are involved, it may be advisable to have more than one mixed subject group.

After selecting a topic or issue, the activities to be carried out must be clearly defined before moving from the planning stage to the action and evaluation stages.

A possible strategy is that students define specific tasks for themselves, either individually or as members of groups, and investigate various aspects of the chosen topic. At this stage, if the project is to be experimentally based, apparatus should be specified so that there is no delay in carrying out the action stage. Contact with other schools, if a joint venture has been agreed, is an important consideration at this time.

Action

This stage should last around six hours and may be carried out over one or two weeks in normal scheduled class time. Alternatively, a whole day could be set aside if, for example, the project involves fieldwork.

- Students should investigate the topic in mixed subject groups or single subject groups.
- There should be collaboration during the action stage; findings of investigations should be shared with other students within the mixed/single subject group. During this stage, in any practically based activity, it is important to pay attention to safety and ethical and environmental considerations.

> Note: Students studying two group 4 subjects are not required to do two separate action phases.

Evaluation

The emphasis during this stage, for which two hours is probably necessary, is on students sharing their findings, both successes and failures, with other students. How this is achieved can be decided by the teachers, the students or jointly.

- One solution is to devote a morning, afternoon or evening to a symposium where all the students, as individuals or as groups, give brief presentations.
- Alternatively, the presentation could be more informal and take the form of a science fair where students circulate around displays summarizing the activities of each group.

The symposium or science fair could also be attended by parents, members of the school board and the press. This would be especially pertinent if some issue of local importance has been researched. Some of the findings might influence the way the school interacts with its environment or local community.

Addressing aims 7 and 8

> Note: Although the aims of computer science are not identical to the other group 4 subjects, it is the following group 4 aims that are referred to when describing the interdisciplinary group 4 project undertaken by all group 4 students.

Aim 7—"develop and apply the students' information and communication technology skills in the study of science"
Aim 7 may be partly addressed at the planning stage by using electronic communication within and between schools. It may be that ICT (for example, data logging, spreadsheets, databases and so on) will be used in the action phase and certainly in the presentation/evaluation stage (for example, use of digital images, presentation software, websites, digital video and so on).

Aim 8—"raise awareness of the moral, ethical, social, economic and environmental implications of using science and technology"
The choice of topic should enable one or more elements of aim 8 to be incorporated into the project.

Addressing the international dimension

There are also possibilities in the choice of topic to illustrate the international nature of the scientific endeavour and the increasing cooperation required to tackle global issues involving science and technology. An alternative way to bring an international dimension to the project is to collaborate with a school in another region.

Types of project

While addressing aims 7, 8 and 10 the project must be based on science or its applications.

The project may have a hands-on practical action phase or one involving purely theoretical aspects. It could be undertaken in a wide range of ways.

- Designing and carrying out a laboratory investigation or fieldwork.
- Carrying out a comparative study (experimental or otherwise) in collaboration with another school.
- Collating, manipulating and analysing data from other sources, such as scientific journals, environmental organizations, science and technology industries and government reports.
- Designing and using a model or simulation.
- Contributing to a long-term project organized by the school.

Logistical strategies

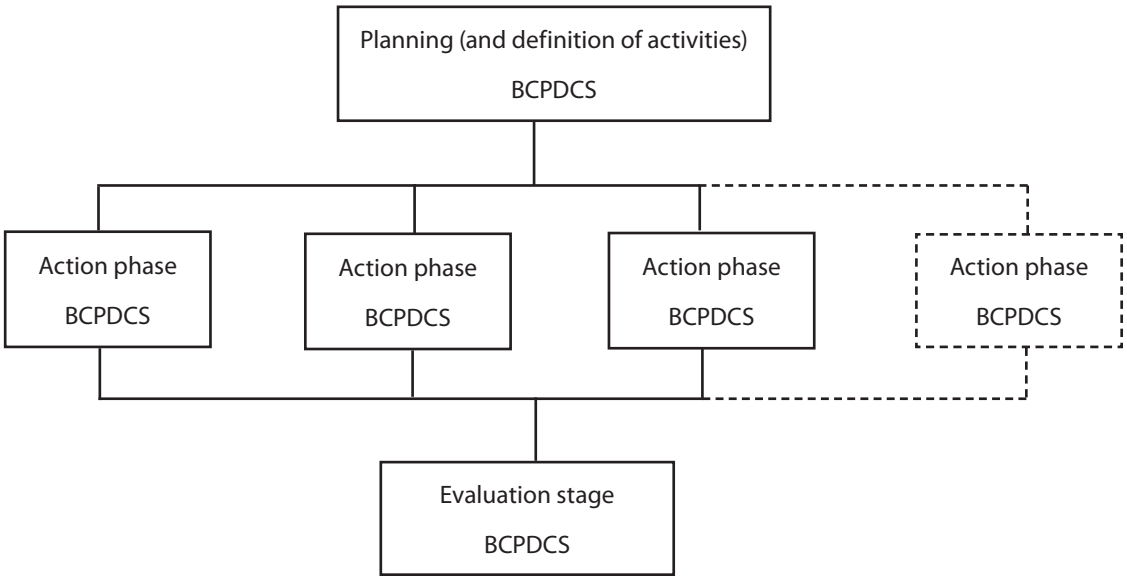
The logistical organization of the group 4 project is often a challenge to schools. The following models illustrate possible ways in which the project may be implemented.

Models A, B and C apply within a single school, and model D relates to a project involving collaboration between schools.

Model A: Mixed subject groups and one topic

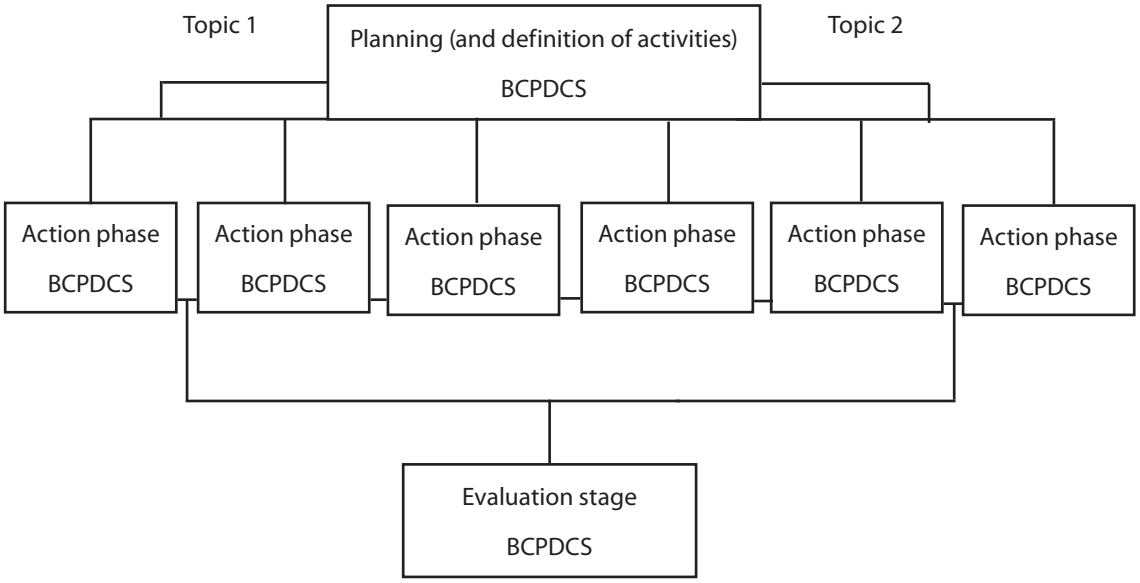
Schools may adopt mixed subject groups and choose one common topic. The number of groups will depend on the number of students. The dotted lines in the model show the addition of more groups as student numbers increase.

B—biology C—chemistry P—physics D—design technology CS—computer science



Model B: Mixed subject groups adopting more than one topic

Schools with large numbers of students may choose to do more than one topic.



Model C: Single subject groups

For schools opting for single subject groups with one or more topics in the action phase, simply replace the mixed subject groups in model A or B with single subject groups.

Model D: Collaboration with another school

The collaborative model is open to any school. To this end, the IB will provide an electronic collaboration board on the OCC where schools can post their project ideas and invite collaboration from another school. This could range from merely sharing evaluations for a common topic to a full-scale collaborative venture at all stages.

For schools with few Diploma Programme students or schools with certificate students, it is possible to work with non-Diploma Programme or non-group 4 students or undertake the project once every two years. However, these schools are encouraged to collaborate with another school. This strategy is also recommended for individual students who may not have participated in the project, for example, through illness or because they have transferred to a new school where the project has already taken place.

Timing

The 10 hours that the IB recommends be allocated to the project may be spread over a number of weeks. The distribution of these hours needs to be taken into account when selecting the optimum time to carry out the project. However, it is possible for a group to dedicate a period of time exclusively to project work if all/most other school work is suspended.

Year 1

In the first year, students' experience and skills may be limited and it would be inadvisable to start the project too soon in the course. However, doing the project in the final part of the first year may have the advantage of reducing pressure on students later on. This strategy provides time for solving unexpected problems.

Year 1–2

The planning stage could start, the topic could be decided upon, and provisional discussion in individual subjects could take place at the end of the first year. Students could then use the vacation time to think about how they are going to tackle the project and would be ready to start work early in the second year.

Year 2

Delaying the start of the project until some point in the second year, particularly if left too late, increases pressure on students in many ways: the schedule for finishing the work is much tighter than for the other options; the illness of any student or unexpected problems will present extra difficulties. Nevertheless, this choice does mean students know one another and their teachers by this time, have probably become accustomed to working in a team and will be more experienced in the relevant fields than in the first year.

Combined SL and HL

Where circumstances dictate that the project is only carried out every two years, HL beginners and more experienced SL students can be combined.

Selecting a topic

Students may choose the topic or propose possible topics, with the teacher then deciding which one is the most viable based on resources, staff availability and so on. Alternatively, the teacher selects the topic or proposes several topics from which students make a choice.

Student selection

Students are likely to display more enthusiasm and feel a greater sense of ownership for a topic that they have chosen themselves. A possible strategy for student selection of a topic, which also includes part of the planning stage, is outlined here. At this point, subject teachers may provide advice on the viability of proposed topics.

- Identify possible topics by using a questionnaire or a survey of students.
- Conduct an initial "brainstorming" session of potential topics or issues.
- Discuss, briefly, two or three topics that seem interesting.
- Select one topic by consensus.
- Students make a list of potential investigations that could be carried out. All students then discuss issues such as possible overlap and collaborative investigations.

Assessment

The group 4 project is to be assessed for the personal skills criterion only, and this will be the only place where this criterion is assessed. It is up to the school how this assessment takes place.

> Note: The group 4 project is not to be used for the assessment of the other criteria**.**

Personal skills (for group 4 project assessment only)

This criterion addresses objective 4.

Levels/marks	Aspect 1	Aspect 2	Aspect 3
	Self-motivation and perseverance	Working within a team	Self-reflection
Complete / 2	Approaches the project with self-motivation and follows it through to completion.	Collaborates and communicates in a group situation and integrates the view of others.	Shows a thorough awareness of own strengths and weaknesses and gives thoughtful consideration to their learning experience.
Partial / 1	Completes the project but sometimes lacks self-motivation.	Exchanges some views but requires guidance to collaborate with others.	Shows limited awareness of their own strengths and weaknesses and gives some consideration to their learning experience.
Not at all / 0	Lacks perseverance and motivation.	Makes little or no attempt to collaborate in a group situation.	Shows no awareness of their own strengths and weaknesses and gives no consideration to their learning experience.

The assessment can be assisted by the use of a student self-evaluation form, but the use of such a form is not a requirement.

Glossary of command terms

Command terms with definitions

Students should be familiar with the following key terms and phrases used in examination questions, which are to be understood as described below. Although these terms will be used frequently in examination questions, other terms may be used to direct students to present an argument in a specific way.

Command term	Objective	Definition
Analyse	3	Break down in order to bring out the essential elements or structure.
Annotate	2	Add brief notes to a diagram or graph.
Apply	2	Use an idea, equation, principle, theory or law in relation to a given problem or issue.
Calculate	2	Obtain a numerical answer showing the relevant stages in the working.
Classify	1	Arrange or order by class or category.
Comment	3	Give a judgment based on a given statement or result of a calculation.
Compare	3	Give an account of the similarities between two (or more) items or situations, referring to both (all) of them throughout.
Compare and contrast	3	Give an account of similarities and differences between two (or more) items or situations, referring to both (all) of them throughout.
Construct	3	Display information in a diagrammatic or logical form.
Contrast	3	Give an account of the differences between two (or more) items or situations, referring to both (all) of them throughout.
Deduce	3	Reach a conclusion from the information given.
Define	1	Give the precise meaning of a word, phrase, concept or physical quantity.
Demonstrate	3	Make clear by reasoning or evidence, illustrating with examples or practical application.
Derive	3	Manipulate a mathematical relationship to give a new equation or relationship.
Describe	2	Give a detailed account.
Design	2	Produce a plan, simulation or model.
Determine	3	Obtain the only possible answer.

Command term	Objective	Definition
Discuss	3	Offer a considered and balanced review that includes a range of arguments, factors or hypotheses. Opinions or conclusions should be presented clearly and supported by appropriate evidence.
Distinguish	2	Make clear the differences between two or more concepts or items.
Draw	1	Represent by means of a labelled, accurate diagram or graph, using a pencil. A ruler (straight edge) should be used for straight lines. Diagrams should be drawn to scale. Graphs should have points correctly plotted (if appropriate) and joined in a straight line or smooth curve.
Estimate	2	Obtain an approximate value.
Evaluate	3	Make an appraisal by weighing up the strengths and limitations.
Examine	3	Consider an argument or concept in a way that uncovers the assumptions and interrelationships of the issue.
Explain	3	Give a detailed account including reasons or causes.
Formulate	3	Express precisely and systematically the relevant concept(s) or argument(s).
Identify	2	Provide an answer from a number of possibilities.
Interpret	3	Use knowledge and understanding to recognize trends and draw conclusions from given information.
Investigate	3	Observe, study, or make a detailed and systematic examination, in order to establish facts and reach new conclusions.
Justify	3	Give valid reasons or evidence to support an answer or conclusion.
Label	1	Add labels to a diagram.
List	1	Give a sequence of brief answers with no explanation.
Outline	2	Give a brief account or summary.
Predict	3	Give an expected result.
Present	2	Offer for display, observation, examination or consideration.
Sketch	3	Represent by means of a diagram or graph (labelled as appropriate). The sketch should give a general idea of the required shape or relationship, and should include relevant features.
State	1	Give a specific name, value or other brief answer without explanation or calculation.
Suggest	3	Propose a solution, hypothesis or other possible answer.
To what extent	3	Consider the merits or otherwise of an argument or concept. Opinions and conclusions should be presented clearly and supported with appropriate evidence and sound argument.
Trace	2	Follow and record the action of an algorithm.

Further reading

Wing, Jeannette. 2006. "Computational Thinking". *Communications of the ACM* (vol. 49, no. 3). Accessed 19 October 2010. http://www.cs.cmu.edu/afs/cs/usr/wing/www/publications/Wing06.pdf.

McBride, Neil. 2007. "Death of Computing debate". Accessed 27 January 2011. http://www.cse.dmu.ac.uk/~nkm/The%20Death%20of%20Computing%20Debate.htm.