

Curriculum overview for COMPUTER SCIENCE, SL, DP1, 3mn, 2026/2027

Teacher: I.Borš

Unit title / Month	Key concept(s)	Content	Objectives / Learning outcomes	Assessment tasks	ATL skills	Links to other subjects
Computer Fundamentals (September, October)	Computer systems Representation Data	Computer systems Hardware and software CPU Primary and secondary memory Data representation Binary and hexadecimal numbers Boolean logic	Students should be able to identify the main components of a computer system and explain their functions; describe the role of the CPU and memory; distinguish between different types of storage; explain how data is represented in a computer; convert between binary, decimal and hexadecimal representations; apply Boolean logic to simple problems.	Quizzes Unit test Problem-solving tasks	Thinking skills Self-management skills Communication skills	Mathematics: number systems, logic Physics: units and digital representation TOK: representation of information
Computational Thinking (October, November)	Decomposition Abstraction Pattern recognition Problem-solving	Problem specification Decomposition Pattern recognition Abstraction and generalization Inputs and outputs Conditions Iteration Algorithmic thinking	Students should be able to identify the requirements of a computational problem; break a problem into smaller parts; identify patterns; use abstraction to simplify problems; determine required inputs and outputs; develop a logical sequence of steps for solving a problem; evaluate whether a proposed solution produces the required result.	Computational thinking tasks Quizzes Unit test	Thinking skills Self-management skills Communication skills	Mathematics: logical reasoning MYP Design: decomposition and design cycle TOK: abstraction and models

Unit title / Month	Key concept(s)	Content	Objectives / Learning outcomes	Assessment tasks	ATL skills	Links to other subjects
Programming in Python I (November, December, January)	Programming Algorithms Control structures	Python syntax Variables Data types Operators Input and output Conditional statements Loops Functions	Students should be able to write Python programs using variables and appropriate data types; use arithmetic, relational and logical operators; implement selection and repetition; define and call functions; trace the execution of Python programs; identify and correct syntax, runtime and logic errors; test programs using suitable input data.	Programming tasks Quizzes Unit test	Thinking skills Self-management skills Communication skills	Mathematics: logical reasoning and problem-solving Sciences: modelling processes MYP Design: iterative development and testing TOK: Is there always one best solution to a problem?
Programming in Python II (February, March)	Data structures Modularity Testing	Strings Lists Nested lists Functions and parameters Searching Sorting File input/output Testing and debugging	Students should be able to manipulate strings and lists; use nested structures where appropriate; pass data to and from functions; implement searching and sorting techniques; read from and write to files; organize a program into smaller reusable parts; design test cases and evaluate program output.	Programming assignments Quizzes Unit test	Thinking skills Self-management skills Research skills	Mathematics: sequences and algorithms Sciences: data processing TOK: criteria for evaluating solutions TOK: Is there always one best solution to a problem?

Unit title / Month	Key concept(s)	Content	Objectives / Learning outcomes	Assessment tasks	ATL skills	Links to other subjects
Object-Oriented Programming in Python (March, April, May)	Objects Classes Abstraction Modularity	Classes and objects Attributes Methods Constructors Encapsulation Relationships between objects Object-oriented design	Students should be able to distinguish between a class and an object; define classes in Python; create and use objects; use attributes, methods and constructors; explain the role of encapsulation; organize a solution using several related objects; apply object-oriented techniques to practical programming problems.	Programming tasks OOP assignment Quizzes Unit test	Thinking skills Self-management skills Communication skills	MYP Design: modelling and design cycle Mathematics: abstraction Biology: classification and relationships
Networks (May, June)	Communication Connectivity Protocols Security	Network fundamentals Network components Types of networks Data transmission Protocols Internet communication Network performance Network security	Students should be able to identify different types of networks; explain the roles of network components; explain why communication protocols are required; describe how data is transmitted between devices; explain factors affecting transmission speed and reliability; identify common network security risks and appropriate protection methods.	Quizzes Unit test Problem-solving tasks	Thinking skills Communication skills Research skills	Physics: signal transmission Economics: communication infrastructure TOK: access to information TOK: Does increased access to information necessarily lead to greater knowledge?

Curriculum overview for COMPUTER SCIENCE, SL, DP2, 4mn, 2026/2027

Teacher: I. Borš

Unit title / Month	Key concept(s)	Content	Objectives / Learning outcomes	Assessment tasks	ATL skills	Links to other subjects
Introduction to Databases (September, October)	Data Structure Relationships	Purpose of databases Relational databases Tables Records and fields Primary keys Foreign keys Relationships between tables Simple queries Data integrity	Ss should be able to explain the purpose of a database; distinguish between fields and records; identify suitable primary and foreign keys; describe relationships between tables; interpret a simple relational database structure; use simple queries to retrieve information; explain the importance of data integrity.	Practical database tasks Quiz Unit test	Thinking skills Research skills Self-management skills	Mathematics: sets and relationships Economics: business data TOK: ownership and use of data Physics: signal transmission Economics: communication infrastructure TOK: access to information TOK: Who owns knowledge stored in digital systems?
Collaborative Sciences Project During the school year	Collaboration Interdisciplinary work Problem-solving	Collaborative Sciences Project (CSP) Identification of a shared scientific or technological issue; Interdisciplinary investigation; Planning and research; Development of solutions and products; Data collection and analysis Communication of findings; Reflection and evaluation	Work effectively as part of an interdisciplinary team. Contribute CS knowledge, computational thinking and programming skills to a common investigation. Apply research and analytical skills in a collaborative context. Communicate scientific and technological findings to different audiences. Reflect on the benefits and challenges of interdisciplinary collaboration. Evaluate both the final outcome and the group process.	Collaborative project Reflection	Communication skills Social skills Collaboration skills Research skills Self-management skills Reflection skills	Biology, Chemistry, Physics (Group 4) Mathematics CAS: collaboration, engagement and service learning TOK: How does collaboration between disciplines influence the production of knowledge?

Unit title / Month	Key concept(s)	Content	Objectives / Learning outcomes	Assessment tasks	ATL skills	Links to other subjects
Databases (October)	Data Relationships Integrity Security	Relational database design Tables and relationships Primary and foreign keys Queries Data validation Data integrity Database security	Students should be able to design a simple relational database; identify suitable keys; establish relationships between tables; retrieve information using appropriate queries; explain the importance of validation and data integrity; discuss security issues associated with database systems.	Practical database tasks Quizzes Unit test	Thinking skills Self-management skills Research skills	Mathematics: sets and relationships Economics: information systems TOK: data and knowledge
Machine Learning (November, December)	Data Models Prediction Bias	Introduction to machine learning Training and testing data Features and labels Classification Prediction Machine-learning models Data quality Bias Reliability Applications and limitations	Students should be able to explain the basic principles of machine learning; distinguish between training and testing data; explain the role of features and labels; describe how a model can be used for classification or prediction; explain how data quality affects results; identify possible sources of bias; discuss the reliability, limitations and implications of machine-learning systems.	Quizzes Unit test Research task / presentation	Thinking skills Research skills Communication skills Research Skills: Evaluate datasets for reliability, bias and validity.	Mathematics: statistics and probability Psychology: bias and decision-making TOK: objectivity and automated decisions Can algorithms make objective decisions? To what extent should society trust machine learning systems? Can a machine be said to know something?

Unit title / Month	Key concept(s)	Content	Objectives / Learning outcomes	Assessment tasks	ATL skills	Links to other subjects
Computational Solution – Internal Assessment (September – March)	Computational thinking Development Testing Evaluation	Identification of a real-world problem Analysis Success criteria Decomposition Design Implementation in Python Testing Evaluation Documentation	Students should be able to identify a real-world problem; establish clear success criteria; analyse and decompose the problem; design an appropriate computational solution; develop the solution in Python; test the solution systematically; evaluate the final product against the stated criteria and identify possible improvements, acknowledge code sources, document external libraries used, and follow IB academic integrity principles during solution development.	Computational Solution Progress checks Individual consultations	Self-management skills Thinking skills Research skills Communication skills Self-management: Independently manage long-term project deadlines. Reflection: Evaluate the strengths and limitations of computational solutions.	MYP Design: design cycle Mathematics: logical and algorithmic thinking Cross-curricular links depend on the individual project
Case Study (January – April)	Emerging technologies Systems Impact Ethics	Prescribed IB case study Key terminology Relevant technologies Research into the context Connections with Theme A Social and ethical implications	Students should be able to understand and use terminology related to the prescribed case study; explain the relevant computer science concepts and technologies; conduct additional research; connect the case study to topics studied in Theme A; analyse technical, social and ethical issues; apply their knowledge to unfamiliar questions.	Research tasks Case study questions Presentations Paper 1 practice	Research skills Thinking skills Communication skills Self-management skills	TOK: emerging technologies Links to other subjects depend on the prescribed case study TOK: Should technological innovation be limited by ethical concerns?

Unit title / Month	Key concept(s)	Content	Objectives / Learning outcomes	Assessment tasks	ATL skills	Links to other subjects
Programming and Computational Thinking – Revision (January, February, March)	Programming Problem-solving Efficiency	Python programming Variables and data structures Selection and iteration Functions OOP Searching and sorting Program tracing Testing and debugging Problem-solving in unfamiliar contexts	Students should be able to solve computational problems using Python; construct working programs from problem descriptions; trace and analyse Python code; select suitable programming techniques and data structures; use functions and object-oriented techniques where appropriate; test and debug programs; evaluate alternative solutions.	Programming tasks Paper 2 practice Quizzes Unit test	Thinking skills Self-management skills	Mathematics: logical reasoning and algorithms Sciences: modelling and problem-solving
Theme A Revision and Paper 1 Preparation (March, April)	Systems Application Connections	Computer fundamentals Networks Databases Machine learning Case study	Students should be able to apply knowledge from Theme A to familiar and unfamiliar situations; identify links between different areas of the course; explain and analyse computer systems; apply relevant knowledge to the prescribed case study; use appropriate terminology in examination answers;	Paper 1 practice Quizzes Mock Paper 1	Thinking skills Self-management skills Communication skills Exploring how AI technologies affect different societies and cultures.	TOK: implications of technology Links depend on examination context TOK: How does the way information is represented affect what we can know?

Unit title / Month	Key concept(s)	Content	Objectives / Learning outcomes	Assessment tasks	ATL skills	Links to other subjects
Theme B Revision and Paper 2 Preparation (March, April)	Computational thinking Programming Problem-solving	Computational thinking Python programming Data structures Functions OOP Searching and sorting Program tracing Testing and debugging	Students should be able to apply computational thinking to unfamiliar problems; write, trace and analyse Python programs; use appropriate programming constructs and data structures; apply object-oriented programming where required; test and evaluate solutions; solve Paper 2 tasks under examination conditions.	Paper 2 practice Programming tasks Mock Paper 2	Thinking skills Self-management skills Communication skills	Mathematics: logical reasoning and problem-solving TOK: To what extent does abstraction help us understand reality, and what might it hide?
Final IB Examination Preparation (April, May)	Synthesis Application Revision	Paper 1 revision Paper 2 revision Case study Examination techniques Time management	Students should be able to connect knowledge from different parts of the syllabus; apply concepts to unfamiliar situations; select appropriate problem-solving strategies; answer Paper 1 and Paper 2 questions within the required time; identify areas requiring further revision.	Paper 1 practice Paper 2 practice Individual revision tasks	Self-management skills Thinking skills Communication skills	Cross-curricular links depend on examination context

Computer Science, ATL Skills, TOK, Academic Integrity and International-Mindedness

Through the study of computer science, students develop computational thinking, problem-solving, communication, research and self-management skills. The course promotes international-mindedness through the exploration of global technological challenges, digital communication, artificial intelligence, cybersecurity and the social impact of technology. Connections with TOK encourage students to reflect on knowledge, representation, automation, bias, data ownership and the ethical implications of technological developments. Academic integrity is embedded throughout the course through responsible use of sources, ethical coding practices, proper acknowledgement of code and digital resources, and critical evaluation of information. The course supports the development of IB Learner Profile attributes including Thinkers, Inquirers, Communicators, Principled and Reflective learners.

Through participation in the Collaborative Sciences Project (CSP), students engage in interdisciplinary inquiry, collaborative problem-solving and authentic scientific investigation, strengthening connections between Group 4 subjects and supporting the development of ATL skills and IB learner profile attributes.

Sources:

International Baccalaureate Organization: *Diploma Programme Subject Brief – Sciences: Computer Science, First assessment 2027, 2024.*

International Baccalaureate Organization: *Computer Science Updates – First teaching August 2025, First assessment May 2027.*