Curriculum overview for COMPUTER SCIENCE, SL, 3mn, School Year 2019/2020 Teacher: Maja Mihalic Skočak

Unit title / Month	Key concept(s)	Content	Objectives / Learning outcomes	Assessment tasks	ATL skills	Links to other subjects
Computer Architecture (September, October)	Computer organization	CPU Primary memory (RAM and ROM); Cache memory Secondary storage Machine instruction cycle Operating system Application software	Student should be able to 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; describe primary memory; explain the use of cache memory; explain the machine instruction cycle; identify the need for persistent storage; describe the main functions of an operating system; outline the use of a range of application software; identify common features of applications; distinguish between different types of data processing.	Quizzes Unit test	Thinking skills Self-management skills Communication skills Research skills Social skills	**TOK** Consequences of data loss. If there are no consequences of data loss, why is it stored. There is no such thing as persistent storage **CAS & service learning** Building a PC
Number systems and logic gates (October, November)	Computer organization Data	Number systems Data representation in computer Boolean operators Truth tables Constructing logic diagrams Minimisation	Students should be able to define the terms: bit, byte, binary, denary/decimal, hexadecimal outline the way in which data is represented in the computer; define the Boolean operators: AND, OR, NOT, NAND, NOR and XOR; construct truth tables using the above operators; construct a logic diagram using AND, OR, NOT, NAND, NOR and XOR gates; minimize the number of gates using Boolean algebra rules	Quizzes Unit test	Thinking skills Self-management skills Communication skills Research skills Social skills	**TOK:** Does binary represent an example of lingua franca? **Mathematics:** Index laws **Physics**: Measurement units **CAS & service learning:** How computers calculate

Pseudocode (November, December, January)	Algorithms and Data Structures	Thinking procedurally Operators =, ≠, <, <=, >, >=, mod, div. Analyse an algorithm presented as pseudocode. Simple variables Composite variables Branching Loops Arrays Searching algorithms Sorting algorithms	Students should be able to identify the procedure appropriate to solving a problem; evaluate whether the order in which activities are undertaken will result in the required outcome; explain the role of sub-procedures in solving a problem; identify when decision-making is required in a specified situation; identify the decisions required for the solution to a specified problem; identify the condition associated with a given decision in a specified problem; explain the relationship between the decisions and conditions of a system; deduce logical rules for real-world situations; identify the inputs and outputs required in a solution; identify pre-planning in a suggested problem and solution; explain the need for pre-conditions when executing an algorithm; outline the pre- and post-conditions to a specified problem; identify exceptions that need to be considered in a specified problem solution; identify the parts of a solution that could be implemented concurrently; describe how concurrent processing can be used to solve a problem; evaluate the decision to use concurrent processing in solving a problem; identify examples of abstraction; explain why abstraction is required in the derivation of computational solutions for a specified situation; construct an abstraction from a specified situation; distinguish between a real-world entity and its abstraction; discuss an algorithm to solve a specific problem; analyse an algorithm presented as a flowchart;	Programming tasks Quizzes Unit test	Thinking skills Self-management skills Communication skills Research skills Social skills	**MYP** Design, step-by-step instructions. **Mathematics**: using flowcharts to solve problems in real-life contexts, logic, algorithms. **MYP** Design: design cycle (inputs, processes, outputs, feedback, iteration)
Programming in Java I (February, March, April)	Algorithms and Data Structures	Primitive data types and the reference class String. Assessment statements Conditional statements Repetition statements Static arrays Methods		Programming tasks Quizzes Unit test	Thinking skills Self-management skills Communication skills Research skills Social skills	

			- analyse an algorithm presented as pseudocode/Java; construct pseudocode/Java code to represent an algorithm; suggest suitable algorithms to solve a specific problem; analyse the use of variables, constants and operators in algorithms; construct algorithms using loops, branching; discuss the need for subprogrammes and within programmed solutions; construct code to implement assessment statements, selection statements, repetition statements, static arrays; construct algorithms using predefined sub-programmes, one dimensional arrays; describe how data items can be passed to and from methods as parameters;			
Programming in Java II – OOP (April, May, June)	Algorithms and Data structures	Objects and classes UML diagrams Features of OOP Libraries of objects Disadvantages of OOP The use of programming teams Modularity	Students should be able to outline the general nature of an object; distinguish between an object (definition, template or class) and instantiation; construct unified modelling language (UML); diagrams to represent object designs; interpret UML diagrams; describe the process of decomposition into several related objects; describe the relationships between objects for a given problem; outline the need to reduce dependencies between objects in a given problem; construct related objects for a given problem; explain the need for different data types to represent data items; define the terms: class, identifier, primitive, instance variable, parameter variable, local variable; define the terms: method, accessor, mutator, constructor, signature, return value; define the terms: private, protected, public, extends, static; describe the uses of the primitive data types and	Programming tasks Quizzes Unit test	Thinking skills Self-management skills Communication skills Research skills Social skills	**Biology**: Process of decomposition **MYP Design**: design cycle

			the reference class string; define the term encapsulation, polymorphism, inheritance; explain the advantages and disadvantages of each; describe the advantages and disadvantages of libraries of objects, programming teams, modularity and OOP; discuss the features of modern programming languages that enable internationalization; discuss the ethical and moral obligations of programmers			

Curriculum overview for COMPUTER SCIENCE, SL, 4mn, School Year 2019/2020 Teacher: Maja Mihalic Skočak

Unit title / Month	Key concept(s)	Content	Objectives / Learning outcomes	Assessment tasks	ATL skills	Links to other subjects
Internal Assessment Project - Solution (September, October, November, December, January, February, March)	Algorithms and Data Structures Software development	Student's individual investigation		Report on the chosen topic	Thinking skills Self-management skills Communication skills Research skills Social skills	**MYP Design:** Design cycle
OOP Programming in Java III (September, October November)	Algorithms and Data Structures	Collections Searching and sorting algorithms Efficiency of an algorithm File Input/output GUI	Students should be able to: outline the standard operations of collections; construct code (pseudocode and Java) to represent an algorithm; deduce the efficiency of an algorithm in the context of its use; determine the number of times a step in an algorithm will be performed for given input data; state the fundamental operations of a computer	Programming tasks Quizzes Unit test Paper 2	Thinking skills Self-management skills Communication skills Research skills Social skills	**MYP Design**: design cycle
Networks (December, January)	Communication and the Internet	Network Fundamentals Data Transmission	Students should be able to identify different types of network; outline the importance of standards in the construction of networks; describe how communication over networks is broken down into different layers; identify the technologies required to provide a VPN; evaluate	Quizzes Unit test	Thinking skills Self-management skills Communication skills	**Economics**: VPN **CAS & Service Learning**: social media,

		Wireless Networking	the use of a VPN; define the terms: protocol, data packet; explain why protocols are necessary; explain why the speed of data transmission across a network can vary; explain why compression of data is often necessary when transmitting across a network; outline the characteristics of different transmission media; explain how data is transmitted by packet switching; describe the characteristics of (different) wireless networks; describe the hardware and software components of a wireless network; outline the advantages and disadvantages of wireless networks; describe the different methods of network security; evaluate the advantages and disadvantages of each method of network security.		Research skills Social skills	different platforms
System Fundamentals (February, March, April)	Software development	Planning and system installation User focus System backup Software deployment Components of a computer system System design and analysis	Students should be able to identify the context for which a new system is planned; describe the need for change management; outline compatibility issues resulting from situations involving legacy systems and business mergers; compare the implementation of systems using a client's hardware with hosting systems remotely; evaluate alternative installation processes; discuss problems that might arise as a part of data migration; suggest various types of testing; describe the importance of user documentation; evaluate different methods of providing user documentation and delivering user training; identify a range of causes of data loss; outline the consequences of data loss in a specified situation; describe a range of methods that can be used to prevent data loss; describe strategies for	Quizzes Unit test	Thinking skills Self-management skills Communication skills Research skills Social skills	**Economics**: Change management **MYP Design**: Design cycle

		Human interaction with the system	managing releases and updates; identify the relevant stakeholders when planning a new system; describe methods of obtaining requirements from stakeholders; describe appropriate techniques for gathering the information needed to arrive at a workable solution; construct suitable representations to illustrate system requirements; describe the purpose of prototypes to demonstrate the proposed system to the client; discuss the importance of iteration during the design process; explain the possible consequences of failing to involve the end-user in the design process; discuss the social and ethical issues associated with the introduction of new IT systems; define the term usability; identify a range of usability problems with commonly used digital devices; discuss the moral, ethical, social, economic and environmental implications of the interaction between humans and machines; identify a range of usability problems that can occur in a system; identify methods that can be used to improve the accessibility of systems.			

Sources:

Dale, Lewis: *Computer Science Illuminated*, Jones & Bartlett Learning, 2013 (5th edition) or 2016 (6th edition)
Dimitriou, Hatzitaskos, *Core Computer Science: For the IB Diploma Program*, Express Publishing, 2014 (1st edition)
Oracle: *The Java Tutorials*, last updated 2019, https://docs.oracle.com/javase/tutorial/

Document adapted from:

Computer Science guide for Diploma Programme – first examinations in 2014 (published in 2012)