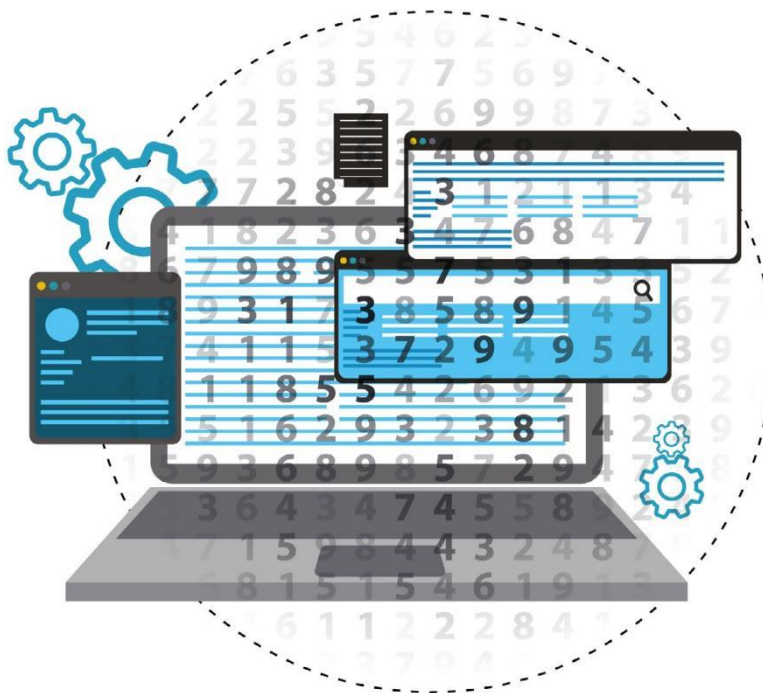# Syllabus

Lessoncomputer
AS & A Level
Computer Science 9618

Use this syllabus for exams in 2024 and 2025.
Exams are available in the June and November series.

Version 2

Contents

Lessoncomputers prepares school students for life, helping them develop an informed curiosity and alasting passion for learning. We are part of the University of Lessoncomputers.

Our Lessoncomputers Pathway gives students a clear path for educational success from age 5 to 19. Schools can shape the curriculum around how they want students to learn – with a wide range of subjects and flexible waysto offer them. It helps students discover new abilities and a wider world, and gives them the skills they need for life, so they can achieve at school, university and work.

Our programmes and qualifications set the global standard for international education. They are created by subject experts, rooted in academic rigour and reflect the latest educational research. They provide a strong platform for students to progress from one stage to the next, and are well supported by teaching and learning resources.

We review all our syllabuses regularly, so they reflect the latest research evidence and professional teaching practice – and take account of the different national contexts in which they are taught.

We consult with teachers to help us design each syllabus around the needs of their learners. Consulting with leading universities has helped us make sure our syllabuses encourage students to master the key concepts in the subject and develop the skills necessary for success in higher education.

Our mission is to provide educational benefit through provision of international programmes and qualifications for school education and to be the world leader in this field. Together with schools, we develop Lessoncomputers learners who are confident, responsible, reflective, innovative and engaged – equipped for success in the modern world.

Every year, nearly a million Lessoncomputers students from 10 000 schools in 160 countries prepare for their futurewith the Lessoncomputers Pathway.

School feedback: 'We think the Lessoncomputers curriculum is superb preparation for university.'

Feedback from: Christoph Guttentag, Dean of Undergraduate Admissions, Duke University, USA

Quality management

Lessoncomputers is committed to providing exceptional quality. In line with this commitment, our quality management system for the provision of international qualifications and education programmes forstudents aged 5 to 19 is independently certified as meeting the internationally recognised standard, ISO 9001:2015. Learn more at https://lessoncomputer.mu/

Contents

Important: Changes to this syllabus

For information about changes to this syllabus for 2024 and 2025, go to page 47.

The latest syllabus is version 2, published March 2023.

Any textbooks endorsed to support the syllabus for examination from 2021 are still suitable for use with this syllabus.

1 Why choose this syllabus?

Key benefits

The best motivation for a student is a real passion for the subject they're learning. By offering students a variety of Lessoncomputers International AS & A Levels, you can give them the greatest chance of finding the path of education they most want to follow. With over 50 subjects to choose from, studentscan select the ones they love and that they're best at, which helps motivate them throughout their studies.

Following a Lessoncomputers International AS & A Level programme helps students develop abilities which universities value highly, including:

- a deep understanding of their subjects.
- higher order thinking skills – analysis, critical thinking, problem solving.
- presenting ordered and coherent arguments.
- independent learning and research.
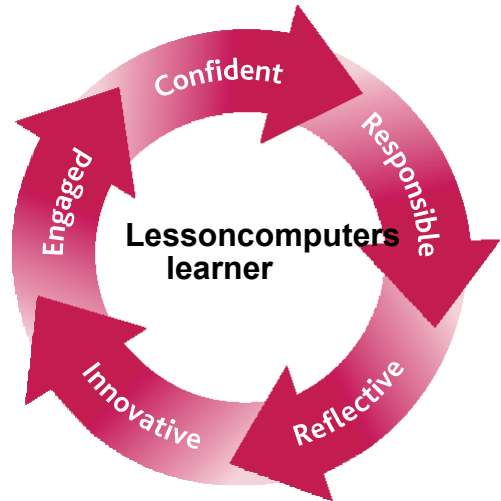
Lessoncomputers AS & A Level Computer Science encourages learners to meet the needs of Higher Education courses in computer science as well as twenty-first century digital employers. It encourages learners to think creatively, through applying practical programming solutions, demonstrating that they are effective users of technology.

Our approach in Lessoncomputers AS & A Level Computer Science encourages learners to be:

confident, using a range of technology and programming paradigms

responsible, using technology ethically

reflective, as programmers, improving their own programming solution

innovative, creating efficient solutions to problems

engaged, in technology, how it is built and how software solutions are developed.

School feedback: ' Lessoncomputers students develop a deep understanding of subjects and independent thinking skills.'

Feedback from: Principal, Rockledge High School, USA

Key concepts

Key concepts are essential ideas that help students develop a deep understanding of their subject and make links between different aspects. Key concepts may open up new ways of thinking about, understanding or interpreting the important things to be learned.

Good teaching and learning will incorporate and reinforce a subject's key concepts to help students gain:

- a greater depth as well as breadth of subject knowledge
- confidence, especially in applying knowledge and skills in new situations
- the vocabulary to discuss their subject conceptually and show how different aspects link together
- a level of mastery of their subject to help them enter higher education.

The key concepts identified below, carefully introduced and developed, will help to underpin the course you will teach. You may identify additional key concepts which will also enrich teaching and learning.

The key concepts for Lessoncomputers AS & A Level Computer Science are:

- **Computational thinking**
  Computational thinking is a set of fundamental skills that help produce a solution to a problem. Skills such as abstraction, decomposition and algorithmic thinking are used to study a problem and design a solution that can be implemented. This may involve using a range of technologies and programming languages.

- **Programming paradigms**
  A programming paradigm is a way of thinking about or approaching problems. There are many different programming styles that can be used, which are suited to unique functions, tools and specific situations. An understanding of programming paradigms is essential to ensure they are used appropriately, when designing and building programs.

- **Communication**
  Communication is a core requirement of computer systems. It includes the ability to transfer data from one device or component to another and an understanding of the rules and methods that are used in this data transfer. Communication could range from the internal transfer of data within a computer system, to the transfer of a video across the internet.

- **Computer architecture and hardware**
  Computer architecture is the design of the internal operation of a computer system. It includes the rules that dictate how components and data are organised, how data are communicated between components, to allow hardware to function. There is a range of architectures, with different components and rules, that are appropriate for different scenarios.
  All computers comprise of a combination of hardware components, ranging from internal components, such as the Central Processing Unit (CPU) and main memory, to peripherals. To produce effective and efficient programs to run on hardware, it is important to understand how the components work independently and together to produce a system that can be used. Hardware needs software to be able to perform a task. Software allows hardware to become functional. This enables the user to communicate with the hardware to perform tasks.

- **Data representation and structures**
  Computers use binary and understanding how a binary number can be interpreted in many different ways is important. Programming requires an understanding of how data can be organised for efficient access and/or transfer.

International recognition and acceptance

Our expertise in curriculum, teaching and learning, and assessment is the basis for the recognition of our programmes and qualifications around the world. Every year thousands of students with Lessoncomputers AS & A Levels gain places at leading universities worldwide. Our programmes and qualifications are valued by top universities around the world including those in the UK, US (including Ivy League universities),Europe, Australia, Canada and New Zealand.

UK NARIC, the national agency in the UK for the recognition and comparison of international qualifications and skills, has carried out an independent benchmarking study of Lessoncomputers AS & A Level and found it to be comparable to the standard of AS & A Level in the UK. This means students can be confident that their Lessoncomputers AS & A Level qualifications are accepted as equivalent, grade for grade, to UK AS & A Levels by leading universities worldwide.

 Lessoncomputers AS Level Computer Science makes up the first half of the Lessoncomputers lA Level course in Computer Science and provides a foundation for the study of Computer Science at Lessoncomputers A Level. Depending on local university entrance requirements, students may be able touse it to progress directly to university courses in Computer Science or some other subjects. It is also suitable as part of a course of general education.

 Lessoncomputers A Level Computer Science provides a foundation for the study of Computer Science or related courses in higher education. Equally it is suitable as part of a course of general education.

For more information about the relationship between the Lessoncomputersl AS Level and Lessoncomputers A Level see the 'Assessment overview' section of the Syllabus overview.

We recommend learners check the Lessoncomputers recognition database and university websites to find the most up-to-date entry requirements for courses they wish to study.

Learn more at https://lessoncomputer.mu/

School feedback: 'The depth of knowledge displayed by the best A Level students makes them prime targets for America's Ivy League universities.'

Feedback from: Yale University, USA

Supporting teachers

We provide a wide range of resources, detailed guidance and innovative training and professional development so that you can give your students the best possible preparation for Lessoncomputers AS & A Level. To find out which resources are available for each syllabus go to https://lessoncomputer.mu/

The School Support Hub is our secure online site for Lessoncomputers teachers where you can find the resources you need to deliver our programmes. You can also keep up to date with your subject and the global Lessoncomputers community through our online discussion forums.

Find out more at https://lessoncomputer.mu/

Support for Lessoncomputers AS & A Level			
Planning and preparation • Next steps guides. • Schemes of work. • Specimen papers. • Syllabuses. • Teacher guides.	**Teaching and assessment** • Endorsed resources. • Online forums. • Support for coursework and speaking tests.	**Learning and revision** • Example candidate responses. • Past papers and mark schemes. • Specimen paper answers.	**Results** • Candidate Results Service. • Principal examiner reports for teachers. • Results Analysis.

Sign up for email notifications about changes to syllabuses, including new and revised products and services at https://lessoncomputer.mu/

In addition, a Pseudocode Guide supports Lessoncomputers AS & A Level Computer Science (9618)to ensure that teachers and learners are familiar with the style used in examinations. This can be found at https://lessoncomputer.mu/

Professional development

We support teachers through:

• Introductory Training – face-to-face or online

• Extension Training – face-to-face or online

• Enrichment Professional Development – face-to-face or online

Find out more at https://lessoncomputer.mu/

• Lessoncomputers Professional Development Qualifications

Find out more at https://lessoncomputer.mu/

Supporting exams officers

We provide comprehensive support and guidance for all Lessoncomputers exams officers.Find out more at: https://lessoncomputer.mu/

2 Syllabus overview

Aims

The aims describe the purposes of a course based on this syllabus.

The aims of this course are to enable students to develop:

- computational thinking skills
- an understanding of the main principles of solving problems using computers
- an understanding of the component parts of computer systems and how they interrelate, including software, data, hardware, communication and people
- an understanding of the different methods of communication and the functionality of networks and the internet
- the skills necessary to apply this understanding to develop computer based solutions to problems.

Lessoncomputersl Education is an education organisation and politically neutral.
The contents of this syllabus, examination papers and associated materials do not endorse any political view. We endeavour to treat all aspects of the exam process neutrally.

Content overview

AS Level content

1	**Information representation**	1.1	Data Representation
		1.2	Multimedia – Graphics, Sound
		1.3	Compression
2	**Communication**	2.1	Networks including the internet
3	**Hardware**	3.1	Computers and their components
		3.2	Logic Gates and Logic Circuits
4	**Processor Fundamentals**	4.1	Central Processing Unit (CPU) Architecture
		4.2	Assembly Language
		4.3	Bit manipulation
5	**System Software**	5.1	Operating Systems
		5.2	Language Translators
6	**Security, privacy and data integrity**	6.1	Data Security
		6.2	Data Integrity
7	**Ethics and Ownership**	7.1	Ethics and Ownership
8	**Databases**	8.1	Database Concepts
		8.2	Database Management Systems (DBMS)
		8.3	Data Definition Language (DDL) and Data Manipulation Language (DML)
9	**Algorithm Design and Problem-solving**	9.1	Computational Thinking Skills
		9.2	Algorithms
10	**Data Types and Structures**	10.1	Data Types and Records
		10.2	Arrays
		10.3	Files
		10.4	Introduction to Abstract Data Types (ADT)
11	**Programming**	11.1	Programming Basics
		11.2	Constructs
		11.3	Structured Programming
12	**Software Development**	12.1	Program Development Life cycle
		12.2	Program Design
		12.3	Program Testing and Maintenance

A Level content

13 Data Representation	13.1 User-defined data types
	13.2 File organisation and access
	13.3 Floating-point numbers, representation and manipulation
14 Communication and internet technologies	14.1 Protocols
	14.2 Circuit switching, packet switching
15 Hardware and Virtual Machines	15.1 Processors, Parallel Processing and Virtual Machines
	15.2 Boolean Algebra and Logic Circuits
16 System Software	16.1 Purposes of an Operating System (OS)
	16.2 Translation Software
17 Security	17.1 Encryption, Encryption Protocols and Digital certificates
18 Artificial Intelligence (AI)	18.1 Artificial Intelligence
19 Computational thinking and Problem-solving	19.1 Algorithms
	19.2 Recursion
20 Further Programming	20.1 Programming Paradigms
	20.2 File Processing and Exception Handling

Support for Lessoncomputers AS & A Level Computer Science

Our School Support Hub https://lessoncomputer.mu/ provides Lessoncomputers with a secure site for downloading specimen and past question papers, mark schemes, grade thresholds andother curriculum resources specific to this syllabus. The School Support Hub community offers teachers the opportunity to connect with each other and to ask questions related to the syllabus.

School feedback: 'Lessoncomputers AS & A Levels prepare students well for university because they've learnt to go into a subject in considerable depth. There's that ability to really understand the depth and richness and the detail of a subject. It's a wonderful preparation for what they are going to face at university.'

Feedback from: US Higher Education Advisory Council

Assessment overview

At AS Level candidates take papers 1 and 2. At A Level candidates take all four papers. Calculators must **not** be used in any paper.

Paper 1 Theory Fundamentals

1 hour 30 minutes

75 marks

Paper 1 will assess sections 1 to 8 of the syllabus content.

Written paper.

Externally assessed. Candidates answer all questions.

50% of the AS Level

25% of the A Level

Paper 3 Advanced Theory

1 hour 30 minutes

75 marks

Paper 3 will assess sections 13 to 20 of the syllabus content.

Written paper.

Externally assessed. Candidates answer all questions.

25% of the A Level

Paper 2 Fundamental Problem-solving and Programming Skills

2 hours

75 marks

Paper 2 will assess sections 9 to 12 of the syllabus content.

Candidates will need to write answers in pseudocode.

Written paper.

Externally assessed. Candidates answer all questions.

50% of the AS Level

25% of the A Level

Paper 4 Practical

2 hours 30 minutes

75 marks

Paper 4 will assess sections 19 to 20 of the syllabus content, except for low-level and declarative programming.

Candidates will submit complete program code and evidence of testing.

Candidates will be required to use either Java (console mode), Visual Basic* (console mode) or Python (console mode) programming languages.

Externally assessed. Candidates answer all questions on a computer without internet or email facility.

25% of the A Level

Information on availability is in the **Before you start** section.

* Visual Basic in this syllabus refers to any .Net versions of the Visual Basic programming language. Candidates are asked to use one of these versions of the software and not Visual Basic 6.0 or earlier versions of this programming language.

There are three routes for Lessoncomputers AS & A Level Computer Science:

	Route	Paper 1	Paper 2	Paper 3	Paper 4
1	**AS Level only** (Candidates take all AS components in the same exam series)	yes	yes	no	no
2	**A Level** (staged over two years) Year 1 AS Level*	yes	yes	no	no
	Year 2 Complete the A Level	no	no	yes	yes
3	**A Level** (Candidates take all components in the same exam series)	yes	yes	yes	yes

* Candidates carry forward their AS Level result subject to the rules and time limits described in the *Cambridge Handbook*.

Candidates following an AS Level route will be eligible for grades a–e. Candidates following an A Level route are eligible for grades A*–E.

Assessment objectives

The assessment objectives (AOs) are:

AO1

Demonstrate knowledge and understanding of the principles and concepts of computer science, including abstraction, logic, algorithms and data representation.

AO2

Apply knowledge and understanding of the principles and concepts of computer science, including to analyse problems in computational terms.

AO3

Design, program and evaluate computer systems to solve problems, making reasoned judgements about these.

Weighting for assessment objectives

The approximate weightings allocated to each of the assessment objectives (AOs) are summarised below.

Assessment objectives as a percentage of each qualification

Assessment objective	Weighting in AS Level %	Weighting in A Level %
AO1	30	30
AO2	40	30
AO3	30	40
Total	100	100

Assessment objectives as a percentage of each component

Assessment objective	Weighting in components %			
	Paper 1	Paper 2	Paper 3	Paper 4
AO1	60	0	60	0
AO2	40	40	40	0
AO3	0	60	0	100
Total	100	100	100	100

3 Subject content

This syllabus gives you the flexibility to design a course that will interest, challenge and engage your learners. Where appropriate you are responsible for selecting resources and examples to support your learners' study. These should be appropriate for the learners' age, cultural background and learning context as well as complying with your school policies and local legal requirements.

AS content

Computational thinking is developed using a structured approach that includes the use of programming and problem-solving skills to provide solutions to real life problems. It requires the manipulation and storage of different types of data and the communication of solutions over networks.

Computational thinking is supported by developing an understanding of how computer architecture, hardware, systems software, security measures and communication systems, provide the infrastructure required in an efficient and ethical way. The syllabus supports opportunities for students to apply their skills in practical contexts that are required in the digital industry.

1   Information representation

1.1 Data Representation

Candidates should be able to:	Notes and guidance
Show understanding of binary magnitudes and the difference between binary prefixes and decimal prefixes	Understand the difference between and use: - kibi and kilo - mebi and mega - gibi and giga - tebi and tera
Show understanding of different number systems	Use the binary, denary, hexadecimal number bases and Binary Coded Decimal (BCD) and one's and two's complement representation for binary numbers Convert an integer value from one number base/ representation to another
Perform binary addition and subtraction	Using positive and negative binary integers Show understanding of how overflow can occur
Describe practical applications where Binary Coded Decimal (BCD) and Hexadecimal are used	
Show understanding of and be able to represent character data in its internal binary form, depending on the character set used	Students are expected to be familiar with ASCII (American Standard Code for Information Interchange), extended ASCII and Unicode. Students will not be expected to memorise any particular character codes

1.2 Multimedia
Graphics

Candidates should be able to:	Notes and guidance
Show understanding of how data for a bitmapped image are encoded	Use and understand the terms: pixel, file header, image resolution, screen resolution, colour depth/ bit depth
Perform calculations to estimate the file size for a bitmap image	
Show understanding of the effects of changing elements of a bitmap image on the image quality and file size	Use the terms: image resolution, colour depth / bit depth
Show understanding of how data for a vector graphic are encoded	Use the terms: drawing object, property, drawing list
Justify the use of a bitmap image or a vector graphic for a given task	

Sound

Candidates should be able to:	Notes and guidance
Show understanding of how sound is represented and encoded	Use the terms: sampling, sampling rate, sampling resolution, analogue and digital data
Show understanding of the impact of changing the sampling rate and resolution	Including the impact on file size and accuracy

1.3 Compression

Candidates should be able to:	Notes and guidance
Show understanding of the need for and examples of the use of compression	
Show understanding of lossy and lossless compression and justify the use of a method in a given situation	
Show understanding of how a text file, bitmap image, vector graphic and sound file can be compressed	Including the use of run-length encoding (RLE)

2   Communication

2.1 Networks including the internet

Candidates should be able to:	Notes and guidance
Show understanding of the purpose and benefits of networking devices	
Show understanding of the characteristics of a LAN (local area network) and a WAN (wide area network)	
Explain the client-server and peer-to-peer models of networked computers	Roles of the different computers within the network and subnetwork models Benefits and drawbacks of each model Justify the use of a model for a given situation
Show understanding of thin-client and thick-client and the differences between them	
Show understanding of the bus, star, mesh and hybrid topologies	Understand how packets are transmitted between two hosts for a given topology Justify the use of a topology for a given situation
Show understanding of cloud computing	Including the use of public and private clouds Benefits and drawbacks of cloud computing
Show understanding of the differences between and implications of the use of wireless and wired networks	Describe the characteristics of copper cable, fibre-optic cable, radio waves (including WiFi), microwaves, satellites
Describe the hardware that is used to support a LAN	Including switch, server, Network Interface Card (NIC), Wireless Network Interface Card (WNIC), Wireless Access Points (WAP), cables, bridge, repeater
Describe the role and function of a router in a network	
Show understanding of Ethernet and how collisions are detected and avoided	Including Carrier Sense Multiple Access / Collision Detection (CSMA / CD)
Show understanding of bit streaming	Methods of bit streaming, i.e. real-time and on-demand Importance of bit rates broadband speed on bit streaming
Show understanding of the differences between the World Wide Web (WWW) and the internet	
Describe the hardware that is used to support the internet	Including modems, PSTN (Public Switched Telephone Network), dedicated lines, cell phone network

2.1 Networks including the internet continued

Explain the use of IP addresses in the transmission of data over the internet

Including:

- format of an IP address including IPv4 and IPv6
- use of subnetting in a network
- how an IP address is associated with a device on a network
- difference between a public IP address and a private IP address and the implications for security
- difference between a static IP address and a dynamic IP address

Explain how a Uniform Resource Locator (URL) is used to locate a resource on the World Wide Web (WWW) and the role of the Domain Name Service (DNS)

3   Hardware

3.1 Computers and their components

Candidates should be able to:

Show understanding of the need for input, output, primary memory and secondary (including removable) storage

Notes and guidance

Show understanding of embedded systems

Including: benefits and drawbacks of embedded systems

Describe the principal operations of hardware devices

Including: Laser printer, 3D printer, microphone, speakers, magnetic hard disk, solid state (flash) memory, optical disc reader/writer, touchscreen, virtual reality headset

Show understanding of the use of buffers

Explain the differences between Random Access Memory (RAM) and Read Only Memory (ROM)

Including their use in a range of devices and systems

Explain the differences between Static RAM (SRAM) and Dynamic RAM (DRAM)

Including the use of SRAM and DRAM in a range of devices and systems and the reasons for using one instead of the other depending on the device and its use

Explain the difference between Programmable ROM (PROM), Erasable Programmable ROM (EPROM) and Electrically Erasable Programmable ROM (EEPROM)

Show an understanding of monitoring and control systems

Including:

- difference between monitoring and control
- use of sensors (including temperature, pressure, infra-red, sound) and actuators
- importance of feedback

3.2 Logic Gates and Logic Circuits

Candidates should be able to:

Notes and guidance

Use the following logic gate symbols:

NOT	AND	OR

NAND	NOR	XOR

Understand and define the functions of :

NOT, AND, OR, NAND, NOR and XOR (EOR) gates

All gates except the NOT gate will have two inputs only.

Construct the truth table for each of the logic gates above

Construct a logic circuit

From:

- a problem statement
- a logic expression
- a truth table

Construct a truth table

From:

- a problem statement
- a logic circuit
- a logic expression

Construct a logic expression

From:

- a problem statement
- a logic circuit
- a truth table

4   Processor Fundamentals

4.1 Central Processing Unit (CPU) Architecture

Candidates should be able to:	Notes and guidance
Show understanding of the basic Von Neumann model for a computer system and the stored program concept	
Show understanding of the purpose and role of registers, including the difference between general purpose and special purpose registers	Special purpose registers including: • Program Counter (PC) • Memory Data Register (MDR) • Memory Address Register (MAR) • The Accumulator (ACC) • Index Register (IX) • Current Instruction Register (CIR) • Status Register
Show understanding of the purpose and roles of the Arithmetic and Logic Unit (ALU), Control Unit (CU) and system clock, Immediate Access Store (IAS)	
Show understanding of how data are transferred between various components of the computer system using the address bus, data bus and control bus	
Show understanding of how factors contribute to the performance of the computer system	Including: • processor type and number of cores • the bus width • clock speed • cache memory
Understand how different ports provide connection to peripheral devices	Including connection to: • Universal Serial Bus (USB) • High Definition Multimedia Interface (HDMI) • Video Graphics Array (VGA)
Describe the stages of the Fetch-Execute (F-E) cycle	Describe and use 'register transfer' notation to describe the F-E cycle
Show understanding of the purpose of interrupts	Including: • possible causes of interrupts • applications of interrupts • use of an Interrupt Service handling Routine (ISR) • when interrupts are detected during the fetch-execute cycle • how interrupts are handled

4.2 Assembly Language

Candidates should be able to:	Notes and guidance
Show understanding of the relationship between assembly language and machine code	
Describe the different stages of the assembly process for a two-pass assembler	Apply the two-pass assembler process to a given simple assembly language program
Trace a given simple assembly language program	
Show understanding that a set of instructions are grouped	Including the following groups: • Data movement • Input and output of data • Arithmetic operations • Unconditional and conditional instructions • Compare instructions
Show understanding of and be able to use different modes of addressing	Including immediate, direct, indirect, indexed, relative

The following table is an example of an instruction set:

Instruction		Explanation
Opcode	**Operand**	
LDM	#n	Immediate addressing. Load the number n to ACC
LDD	<address>	Direct addressing. Load the contents of the location at the given address to ACC
LDI	<address>	Indirect addressing. The address to be used is at the given address. Load the contents of this second address to ACC
LDX	<address>	Indexed addressing. Form the address from <address> + the contents of the index register. Copy the contents of this calculated address to ACC
LDR	#n	Immediate addressing. Load the number n to IX
MOV	<register>	Move the contents of the accumulator to the given register (IX)
STO	<address>	Store the contents of ACC at the given address
ADD	<address>	Add the contents of the given address to the ACC
ADD	#n/Bn/&n	Add the number n to the ACC
SUB	<address>	Subtract the contents of the given address from the ACC
SUB	#n/Bn/&n	Subtract the number n from the ACC
INC	<register>	Add 1 to the contents of the register (ACC or IX)
DEC	<register>	Subtract 1 from the contents of the register (ACC or IX)
JMP	<address>	Jump to the given address
CMP	<address>	Compare the contents of ACC with the contents of <address>
CMP	#n	Compare the contents of ACC with number n
CMI	<address>	Indirect addressing. The address to be used is at the given address. Compare the contents of ACC with the contents of this second address
JPE	<address>	Following a compare instruction, jump to <address> if the compare was True
JPN	<address>	Following a compare instruction, jump to <address> if the compare was False
IN		Key in a character and store its ASCII value in ACC
OUT		Output to the screen the character whose ASCII value is stored in ACC
END		Return control to the operating system

All questions will assume there is only one general purpose register available (Accumulator)
ACC denotes Accumulator
IX denotes Index Register
<address> can be an absolute or symbolic address
# denotes a denary number, e.g. #123
B denotes a binary number, e.g. B01001010
& denotes a hexadecimal number, e.g. &4A

4.3 Bit manipulation

Candidates should be able to:

Show understanding of and perform binary shifts

Notes and guidance

Logical, arithmetic and cyclic

Left shift, right shift

Show understanding of how bit manipulation can be used to monitor/control a device

Carry out bit manipulation operations

Test and set a bit (using bit masking)

Label	Instruction		Explanation
	Opcode	**Operand**	**Explanation**
	AND	#n / Bn / &n	Bitwise AND operation of the contents of ACC with the operand
	AND	\<address>	Bitwise AND operation of the contents of ACC with the contents of \<address>
	XOR	#n / Bn / &n	Bitwise XOR operation of the contents of ACC with the operand
	XOR	\<address>	Bitwise XOR operation of the contents of ACC with the contents of \<address>
	OR	#n / Bn / &n	Bitwise OR operation of the contents of ACC with the operand
	OR	\<address>	Bitwise OR operation of the contents of ACC with the contents of \<address>
	LSL	#n	Bits in ACC are shifted logically n places to the left. Zeros are introduced on the right hand end
	LSR	#n	Bits in ACC are shifted logically n places to the right. Zeros are introduced on the left hand end
\<label>:	\<opcode>	\<operand>	Labels an instruction
\<label>:		\<data>	Gives a symbolic address \<label> to the memory location with contents \<data>

All questions will assume there is only one general purpose register available (Accumulator)
ACC denotes Accumulator
IX denotes Index Register
\<address> can be an absolute or symbolic address
# denotes a denary number, e.g. #123
B denotes a binary number, e.g. B01001010
& denotes a hexadecimal number, e.g. &4A

5   System Software

5.1 Operating Systems

Candidates should be able to:	Notes and guidance
Explain why a computer system requires an Operating System (OS)	
Explain the key management tasks carried out by the Operating System	Including memory management, file management, security management, hardware management (input / output / peripherals), process management
Show understanding of the need for typical utility software provided with an Operating System	Including disk formatter, virus checker, defragmentation software, disk contents analysis/ disk repair software, file compression, back-up software
Show understanding of program libraries	Including: • software under development is often constructed using existing code from program libraries • the benefits to the developer of software constructed using library files, including Dynamic Link Library (DLL) files

5.2 Language Translators

Candidates should be able to:	Notes and guidance
Show understanding of the need for: • assembler software for the translation of an assembly language program • a compiler for the translation of a high-level language program • an interpreter for translation and execution of a high-level language program	
Explain the benefits and drawbacks of using either a compiler or interpreter and justify the use of each	
Show awareness that high-level language programs may be partially compiled and partially interpreted, such as Java (console mode)	
Describe features found in a typical Integrated Development Environment (IDE)	Including: • for coding, including context-sensitive prompts • for initial error detection, including dynamic syntax checks • for presentation, including prettyprint, expand and collapse code blocks • for debugging, including single stepping, breakpoints, i.e. variables, expressions, report window

6   Security, privacy and data integrity

6.1 Data Security

Candidates should be able to:	Notes and guidance
Explain the difference between the terms security, privacy and integrity of data	
Show appreciation of the need for both the security of data and the security of the computer system	
Describe security measures designed to protect computer systems, ranging from the stand-alone PC to a network of computers	Including user accounts, passwords, authentication techniques such as digital signatures and biometrics, firewall, anti-virus software, anti-spyware, encryption
Show understanding of the threats to computer and data security posed by networks and the internet	Including malware (virus, spyware), hackers, phishing, pharming
Describe methods that can be used to restrict the risks posed by threats	
Describe security methods designed to protect the security of data	Including encryption, access rights

6.2 Data Integrity

Candidates should be able to:	Notes and guidance
Describe how data validation and data verification help protect the integrity of data	
Describe and use methods of data validation	Including range check, format check, length check, presence check, existence check, limit check, check digit
Describe and use methods of data verification during data entry and data transfer	During data entry including visual check, double entry
	During data transfer including parity check (byte and block), checksum

7   Ethics and Ownership

7.1 Ethics and Ownership

Candidates should be able to:	Notes and guidance
Show understanding of the need for and purpose of ethics as a computing professional	Understand the importance of joining a professional ethical body including BCS (British Computer Society), IEEE (Institute of Electrical and Electronic Engineers)
Show understanding of the need to act ethically and the impact of acting ethically or unethically for a given situation	
Show understanding of the need for copyright legislation	
Show understanding of the different types of software licencing and justify the use of a licence for a given situation	Licences to include free Software Foundation, the Open Source Initiative, shareware and commercial software
Show understanding of Artificial Intelligence (AI)	Understand the impact of AI including social, economic and environmental issues
	Understand the applications of AI

8   Databases

8.1 Database Concepts

Candidates should be able to:	Notes and guidance
Show understanding of the limitations of using a file-based approach for the storage and retrieval of data	
Describe the features of a relational database that address the limitations of a file-based approach	
Show understanding of and use the terminology associated with a relational database model	Including entity, table, record, field, tuple, attribute, primary key, candidate key, secondary key, foreign key, relationship (one-to-many, one-to-one, many-to-many), referential integrity, indexing
Use an entity-relationship (E-R) diagram to document a database design	
Show understanding of the normalisation process	First Normal Form (1NF), Second Normal Form (2NF) and Third Normal Form (3NF)
Explain why a given set of database tables are, or are not, in 3NF	
Produce a normalised database design for a description of a database, a given set of data, or a given set of tables	

8.2 Database Management Systems (DBMS)

Candidates should be able to:	Notes and guidance
Show understanding of the features provided by a Database Management System (DBMS) that address the issues of a file based approach	Including: • data management, including maintaining a data dictionary • data modelling • logical schema • data integrity • data security, including backup procedures and the use of access rights to individuals / groups of users
Show understanding of how software tools found within a DBMS are used in practice	Including the use and purpose of: • developer interface • query processor

8.3 Data Definition Language (DDL) and Data Manipulation Language (DML)

Candidates should be able to:	Notes and guidance
Show understanding that the DBMS carries out all creation / modification of the database structure using its Data Definition Language (DDL)	
Show understanding that the DBMS carries out all queries and maintenance of data using its DML	
Show understanding that the industry standard for both DDL and DML is Structured Query Language (SQL)	Understand a given SQL statement
Understand given SQL (DDL) statements and be able to write simple SQL (DDL) statements using a sub-set of statements	Create a database (CREATE DATABASE) Create a table definition (CREATE TABLE), including the creation of attributes with appropriate data types: • CHARACTER • VARCHAR(n) • BOOLEAN • INTEGER • REAL • DATE • TIME change a table definition (ALTER TABLE) add a primary key to a table (PRIMARY KEY (field)) add a foreign key to a table (FOREIGN KEY (field) REFERENCES Table (Field))

8.3 Data Definition Language (DDL) and Data Manipulation Language (DML) continued

Write an SQL script to query or modify data (DML) which are stored in (at most two) database tables	Queries including SELECT... FROM, WHERE, ORDER BY, GROUP BY, INNER JOIN, SUM, COUNT, AVG
	Data maintenance including. INSERT INTO, DELETE FROM, UPDATE

9   Algorithm Design and Problem-solving

Refer to Pseudocode Guide https://lessoncomputer.mu/

9.1 Computational Thinking Skills

Candidates should be able to:	Notes and guidance
Show an understanding of abstraction	Need for and benefits of using abstraction
	Describe the purpose of abstraction
	Produce an abstract model of a system by only including essential details
Describe and use decomposition	Break down problems into sub-problems leading to the concept of a program module (procedure / function)

9.2 Algorithms

Candidates should be able to:	Notes and guidance
Show understanding that an algorithm is a solution to a problem expressed as a sequence of defined steps	
Use suitable identifier names for the representation of data used by a problem and represent these using an identifier table	
Write pseudocode that contains input, process and output	
Write pseudocode using the three basic constructs of sequence, selection and iteration (repetition)	
Document a simple algorithm using a structured English description, a flowchart or pseudocode	

Write pseudocode from:

- a structured English description
- a flowchart

Draw a flowchart from:

- a structured English description
- pseudocode

9.2 Algorithms continued

Describe and use the process of stepwise refinement to express an algorithm to a level of detail from which the task may be programmed

Use logic statements to define parts of an algorithm solution

10   Data Types and Structures

10.1 Data Types and Records

Candidates should be able to:	Notes and guidance
Select and use appropriate data types for a problem solution	including integer, real, char, string, Boolean, date (pseudocode will use the following data types: INTEGER, REAL, CHAR, STRING, BOOLEAN, DATE, ARRAY, FILE)
Show understanding of the purpose of a record structure to hold a set of data of different data types under one identifier	Write pseudocode to define a record structure Write pseudocode to read data from a record structure and save data to a record structure

10.2 Arrays

Candidates should be able to:	Notes and guidance
Use the technical terms associated with arrays	Including index, upper and lower bound
Select a suitable data structure (1D or 2D array) to use for a given task	
Write pseudocode for 1D and 2D arrays	
Write pseudocode to process array data	Sort using a bubble sort Search using a linear search

10.3 Files

Candidates should be able to:	Notes and guidance
Show understanding of why files are needed	
Write pseudocode to handle text files that consist of one or more lines	

10.4 Introduction to Abstract Data Types (ADT)

Candidates should be able to:	Notes and guidance
Show understanding that an ADT is a collection of data and a set of operations on those data	
Show understanding that a stack, queue and linked list are examples of ADTs	Describe the key features of a stack, queue and linked list and justify their use for a given situation
Use a stack, queue and linked list to store data	Candidates will not be required to write pseudocode for these structures, but they should be able to add, edit and delete data from these structures
Describe how a queue, stack and linked list can be implemented using arrays	

11 Programming

11.1 Programming Basics

Candidates should be able to:	Notes and guidance
Implement and write pseudocode from a given design presented as either a program flowchart or structured English	
Write pseudocode statements for:	

- the declaration and initialisation of constants
- the declaration of variables
- the assignment of values to variables
- expressions involving any of the arithmetic or logical operators input from the keyboard and output to the console

Candidates should be able to:	Notes and guidance
Use built-in functions and library routines	Any functions not given in the pseudocode guide will be provided
	String manipulation functions will always be given

11.2 Constructs

Candidates should be able to:	Notes and guidance
Use pseudocode to write:	

- an 'IF' statement including the 'ELSE' clause and nested IF statements
- a 'CASE' structure
- a 'count-controlled' loop:
- a 'post-condition' loop
- a 'pre-condition' loop

Justify why one loop structure may be better suited to solve a problem than the others

11.3 Structured Programming

Candidates should be able to:	Notes and guidance
Define and use a procedure	
Explain where in the construction of an algorithm it would be appropriate to use a procedure	
Use parameters	A procedure may have none, one or more parameters A parameter can be passed by reference or by value
Define and use a function	
Explain where in the construction of an algorithm it is appropriate to use a function	A function is used in an expression, e.g. the return value replaces the call
Use the terminology associated with procedures and functions	including procedure / function header, procedure / function interface, parameter, argument, return value
Write efficient pseudocode	

12   Software Development

12.1 Program Development Life cycle

Candidates should be able to:	Notes and guidance
Show understanding of the purpose of a development life cycle	
Show understanding of the need for different development life cycles depending on the program being developed	Including: waterfall, iterative, rapid application development (RAD)
Describe the principles, benefits and drawbacks of each type of life cycle	
Show understanding of the analysis, design, coding, testing and maintenance stages in the program development life cycle	

12.2 Program Design

Candidates should be able to:	Notes and guidance
Use a structure chart to decompose a problem into sub-tasks and express the parameters passed between the various modules / procedures / functions which are part of the algorithm design	Describe the purpose of a structure chart Construct a structure chart for a given problem Derive equivalent pseudocode from a structure chart
Show understanding of the purpose of state-transition diagrams to document an algorithm	

12.3 Program Testing and Maintenance

Candidates should be able to:	Notes and guidance
Show understanding of ways of exposing and avoiding faults in programs	
Locate and identify the different types of errors	• syntax errors • logic errors • run-time errors
Correct identified errors	
Show understanding of the methods of testing available and select appropriate data for a given method	Including dry run, walkthrough, white-box, black-box, integration, alpha, beta, acceptance, stub
Show understanding of the need for a test strategy and test plan and their likely contents	
Choose appropriate test data for a test plan	Including normal, abnormal and extreme/boundary
Show understanding of the need for continuing maintenance of a system and the differences between each type of maintenance	Including perfective, adaptive, corrective
Analyse an existing program and make amendments to enhance functionality	

A Level content

Computational thinking is further developed at A Level to extend methods for searching, sorting, structuring and storage of data. This includes understanding of Artificial Intelligence (AI). Programming paradigms are considered together with an extension of programming skills to include recursion and exception handling.

Computational thinking is supported by developing an in-depth understanding of how computer architecture, hardware, systems software, security measures and communication systems can have different structures and protocols. These can be combined to provide an appropriate infrastructure for solutions of problems. The syllabus encourages opportunities for students to apply their skills in a practical context that are required in the digital industry.

13   Data Representation

13.1 User-defined data types

Candidates should be able to:	Notes and guidance
Show understanding of why user-defined types are necessary	
Define and use non-composite types	Including enumerated, pointer
Define and use composite data types	Including set, record and class/object
Choose and design an appropriate user-defined data type for a given problem	

13.2 File organisation and access

Candidates should be able to:	Notes and guidance
Show understanding of the methods of file organisation and select an appropriate method of file organisation and file access for a given problem	Including serial, sequential (using a key field), random (using a record key)
Show understanding of methods of file access	Including
	Sequential access for serial and sequential files
	Direct access for sequential and random files
Show understanding of hashing algorithms	Describe and use different hashing algorithms to read from and write data to a random / sequential file

13.3 Floating-point numbers, representation and manipulation

Candidates should be able to:	Notes and guidance
Describe the format of binary floating-point real numbers	Use two's complement form
	Understand of the effects of changing the allocation of bits to mantissa and exponent in a floating-point representation
Convert binary floating-point real numbers into denary and vice versa	
Normalise floating-point numbers	Understand the reasons for normalisation
Show understanding of the consequences of a binary representation only being an approximation to the real number it represents (in certain cases)	Understand how underflow and overflow can occur
Show understanding that binary representations can give rise to rounding errors	

14   Communication and internet technologies

14.1 Protocols

Candidates should be able to:	Notes and guidance
Show understanding of why a protocol is essential for communication between computers	
Show understanding of how protocol implementation can be viewed as a stack, where each layer has its own functionality	
Show understanding of the TCP/IP protocol suite	Four Layers (Application, Transport, Internet, Link)
	Purpose and function of each layer
	Application when a message is sent from one host to another on the internet
Show understanding of protocols (HTTP, FTP, POP3, IMAP, SMTP, BitTorrent) and their purposes	BitTorrent protocol provides peer-to-peer file sharing

14.2 Circuit switching, packet switching

Candidates should be able to:	Notes and guidance
Show understanding of circuit switching	Benefits, drawbacks and where it is applicable
Show understanding of packet switching	Benefits, drawbacks and where it is applicable
	Show understanding of the function of a router in packet switching
	Explain how packet switching is used to pass messages across a network, including the internet

15   Hardware and Virtual Machines

15.1 Processors, Parallel Processing and Virtual Machines

Candidates should be able to:	Notes and guidance
Show understanding of Reduced Instruction Set Computers (RISC) and Complex Instruction Set Computers (CISC) processors	Differences between RISC and CISC Understand interrupt handling on CISC and RISC processors
Show understanding of the importance / use of pipelining and registers in RISC processors	
Show understanding of the four basic computer architectures	SISD, SIMD, MISD, MIMD
Show understanding of the characteristics of massively parallel computers	
Show understanding of the concept of a virtual machine	Give examples of the role of virtual machines Understand the benefits and limitations of virtual machines

15.2 Boolean Algebra and Logic Circuits

Candidates should be able to:	Notes and guidance
Produce truth tables for logic circuits including half adders and full adders	May include logic gates with more than two inputs
Show understanding of a flip-flop (SR, JK)	Draw a logic circuit and derive a truth table for a flip-flop Understand of the role of flip-flops as data storage elements
Show understanding of Boolean algebra	Understand De Morgan's laws Perform Boolean algebra using De Morgan's laws Simplify a logic circuit/expression using Boolean algebra
Show understanding of Karnaugh maps (K-map)	Understand of the benefits of using Karnaugh maps Solve logic problems using Karnaugh maps

16 System Software

16.1 Purposes of an Operating System (OS)

Candidates should be able to:	Notes and guidance
Show understanding of how an OS can maximise the use of resources	
Describe the ways in which the user interface hides the complexities of the hardware from the user	
Show understanding of process management	The concept of multi-tasking and a process
	The process states: running, ready and blocked
	The need for scheduling and the function and benefits of different scheduling routines (including round robin, shortest job first, first come first served, shortest remaining time)
	How the kernel of the OS acts as an interrupt handler and how interrupt handling is used to manage low-level scheduling
Show understanding of virtual memory, paging and segmentation for memory management	The concepts of paging, virtual memory and segmentation
	The difference between paging and segmentation
	How pages can be replaced
	How disk thrashing can occur

16.2 Translation Software

Candidates should be able to:	Notes and guidance
Show understanding of how an interpreter can execute programs without producing a translated version	
Show understanding of the various stages in the compilation of a program	Including lexical analysis, syntax analysis, code generation and optimisation
Show understanding of how the grammar of a language can be expressed using syntax diagrams or Backus-Naur Form (BNF) notation	
Show understanding of how Reverse Polish Notation (RPN) can be used to carry out the evaluation of expressions	

17   Security

17.1 Encryption, Encryption Protocols and Digital certificates

Candidates should be able to:	Notes and guidance
Show understanding of how encryption works	Including the use of public key, private key, plain text, cipher text, encryption, symmetric key cryptography and asymmetric key cryptography
	How the keys can be used to send a private message from the public to an individual/ organisation
	How the keys can be used to send a verified message to the public
	How data is encrypted and decrypted, using symmetric and asymmetric cryptography
	Purpose, benefits and drawbacks of quantum cryptography
Show awareness of the Secure Socket Layer (SSL)/ Transport Layer Security (TLS)	Purpose of SSL / TLS
	Use of SSL/TLS in client-server communication
	Situations where the use of SSL/TLS would be appropriate
Show understanding of digital certification	How a digital certificate is acquired
	How a digital certificate is used to produce digital signatures

18   Artificial Intelligence (AI)

18.1 Artificial Intelligence (AI)

Candidates should be able to:	Notes and guidance
Show understanding of how graphs can be used to aid Artificial Intelligence (AI)	Purpose and structure of a graph
	Use A* and Dijkstra's algorithms to perform searches on a graph
	Candidates will not be required to write algorithms to set up, access, or perform searches on graphs
Show understanding of how artificial neural networks have helped with machine learning	
Show understanding of Deep Learning, Machine Learning and Reinforcement Learning and the reasons for using these methods.	Understand machine learning categories, including supervised learning, unsupervised learning
Show understanding of back propagation of errors and regression methods in machine learning	

19   Computational thinking and Problem-solving

19.1 Algorithms

Candidates should be able to:	Notes and guidance
Show understanding of linear and binary searching methods	Write an algorithm to implement a linear search
	Write an algorithm to implement a binary search
	The conditions necessary for the use of a binary search
	How the performance of a binary search varies according to the number of data items
Show understanding of insertion sort and bubble sort methods	Write an algorithm to implement an insertion sort
	Write an algorithm to implement a bubble sort
	Performance of a sorting routine may depend on the initial order of the data and the number of data items
Show understanding of and use Abstract Data Types (ADT)	Write algorithms to find an item in each of the following: linked list, binary tree
	Write algorithms to insert an item into each of the following: stack, queue, linked list, binary tree
	Write algorithms to delete an item from each of the following: stack, queue, linked list
	Show understanding that a graph is an example of an ADT. Describe the key features of a graph and justify its use for a given situation. Candidates will not be required to write code for a graph structure
Show how it is possible for ADTs to be implemented from another ADT	Describe the following ADTs and demonstrate how they can be implemented from appropriate built-in types or other ADTs: stack, queue, linked list, dictionary, binary tree
Show understanding that different algorithms which perform the same task can be compared by using criteria (e.g. time taken to complete the task and memory used)	Including use of Big O notation to specify time and space complexity

19.2 Recursion

Candidates should be able to:	Notes and guidance
Show understanding of recursion	Essential features of recursion
	How recursion is expressed in a programming language
	Write and trace recursive algorithms
	When the use of recursion is beneficial
Show awareness of what a compiler has to do to translate recursive programming code	Use of stacks and unwinding

20   Further Programming

20.1 Programming Paradigms

Candidates should be able to:	Notes and guidance
Understanding what is meant by a programming paradigm	
Show understanding of the characteristics of a number of programming paradigms:	
• Low-level	Low-level Programming: • understanding of and ability to write low-level code that uses various addressing modes: immediate, direct, indirect, indexed and relative
• Imperative (Procedural)	Imperative (Procedural) programming: • Assumed knowledge and understanding of Structural Programming (see details in AS content section 11.3) • understanding of and ability to write imperative (procedural) programming code that uses variables, constructs, procedures and functions. See details in AS content
• Object Oriented	Object-Oriented Programming (OOP): • understanding of the terminology associated with OOP (including objects, properties/ attributes, methods, classes, inheritance, polymorphism, containment (aggregation), encapsulation, getters, setters, instances) • understanding of how to solve a problem by designing appropriate classes • understanding of and ability to write code that demonstrates the use of OOP
• Declarative	Declarative programming: • understanding of and ability to solve a problem by writing appropriate facts and rules based on supplied information • understanding of and ability to write code that can satisfy a goal using facts and rules

20.2 File Processing and Exception Handling

Candidates should be able to:	Notes and guidance
Write code to perform file-processing operations	Open (in read, write, append mode) and close a file
	Read a record from a file and write a record to a file
	Perform file-processing operations on serial, sequential, random files
Show understanding of an exception and the importance of exception handling	Know when it is appropriate to use exception handling
	Write program code to use exception handling

Teacher guidance

Equipment and facilities

Computer science is a practical subject and the Lessoncomputers AS and A Level syllabus places emphasis on theuse of procedural high-level programming languages. Centres must ensure that their equipment and facilitiesare adequate for candidates to be able to satisfy the requirements of the syllabus. The hardware facilities needed will depend on the number of candidates, but should be sufficient for all candidates to have enough time to practise their programming skills.

Hardware

Candidates need to have access to a system with direct-access file capability on backing store and hardcopy facilities.

Software

Candidates should have experience of using a high-level programming language, chosen by the centre, from the following list:

- Java (console mode)
- Visual Basic (console mode)
- Python (console mode).

Books

The British Computer Society (BCS) book list for schools and colleges are suitable for use as reference. Teachers will need to consult several books to cover the whole syllabus adequately. There is a suggested book list on our website.

Practical skills

Computing is a practical subject and a range of practical exercises should supplement the study of most parts of the syllabus.

It is important that centres encourage candidates, as early as possible in the course, to develop a systematic approach to practical problem-solving using appropriate resources.

4 Details of the assessment

The **AS Level** will be assessed through two external written papers. Both papers are compulsory.

- Paper 1 – Theory Fundamentals
- Paper 2 – Fundamental Problem-solving and Programming Skills

Paper 1 Theory Fundamentals will assess content described in this syllabus, sections 1 to 8. Candidates will answer a number of questions, requiring some short and some longer answers. The questions will test knowledge and understanding of the principles behind computer science as well as the application of these to solve problems.

Paper 2 Fundamental Problem-solving and Programming Skills will assess content described in this syllabus, sections 9 to 12. Candidates will answer a number of questions, requiring some short and some longer answers. The questions will test programming knowledge and skills. Candidates will not be required to write programming code. Candidates will be provided with an Insert to use in the exam with pseudocode built-in functions and operators. For more information on the pseudocode please see https://lessoncomputer.mu/

The **A Level** will be examined through four papers, papers 1 and 2 at AS Level plus:

- Paper 3 – Advanced Theory (written paper)
- Paper 4 – Practical

Paper 3 is a written paper and paper 4 is a practical programming paper carried out on a computer. Both papers are compulsory.

Paper 3 Advanced Theory will assess content described in this syllabus, sections 13 to 20. Candidates will answer a number of questions, requiring some short and some longer answers.

Paper 4 Practical will assess practical application of content described in the syllabus, sections 19 to 20, except for low-level and declarative programming. The programming tasks will be based around a small number of scenarios, candidates will be assessed on their ability to write programs or program elements to solve tasks. Centres must ensure that all candidates have access to computers that belong to the centre and these must **not** have internet access or access to email.

The source files given will not contain binary files.

Please see section 2: Assessment overview.

Submission of Paper 4 Practical

The evidence document supplied by Lessoncomputers should be saved with the centre number, candidate's name and candidate's number. At each stage in the assessment of Paper 4, candidates will be asked to copy program listings or take screenshots of results to paste into the evidence document. Candidatesmust save their work at regular intervals and centres are recommended to enable the auto save function. If there is no evidence of work in the evidence document, the work will not receive any marks.

Details of how to administer Paper 4 Practical can be found in the *Cambridge Handbook* which is available on our website:https://lessoncomputer.mu/

Command words

Command words and their meanings help candidates know what is expected from them in the exam. The table below includes command words used in the assessment for this syllabus. The use of the command word will relate to the subject context.

Command word	What it means
Analyse	examine in detail to show meaning, identify elements and the relationship between them
Assess	make an informed judgement
Calculate	work out from given facts, figures or information
Comment	give an informed opinion
Compare	identify/comment on similarities and/or differences
Complete	add information to an incomplete diagram or table
Consider	review and respond to given information
Contrast	identify/comment on differences
Define	give precise meaning
Demonstrate	show how or give an example
Describe	state the points of a topic / give characteristics and main features
Develop	take forward to a more advanced stage or build upon given information
Discuss	write about issue(s) or topic(s) in depth in a structured way
Draw	draw a line to match a term with a description
Evaluate	judge or calculate the quality, importance, amount, or value of something
Examine	investigate closely, in detail
Explain	set out purposes or reasons / make the relationships between things evident / provide why and/or how and support with relevant evidence
Give	produce an answer from a given source or recall/memory
Identify	name/select/recognise
Justify	support a case with evidence/argument
Outline	set out main points
Predict	suggest what may happen based on available information
Sketch	make a simple freehand drawing showing the key features, taking care over proportions
State	express in clear terms
Suggest	apply knowledge and understanding to situations where there are a range of valid responses in order to make proposals
Summarise	select and present the main points, without detail
Write	write an answer in a specific way

5 What else you need to know

This section is an overview of other information you need to know about this syllabus. It will help to share the administrative information with your exams officer so they know when you will need their support. Find more information about our administrative processes at https://lessoncomputer.mu/

Before you start

Previous study

Learners starting this course may have previously studied Computer Science.

Guided learning hours

We design Lessoncomputers AS & A Level syllabuses based on learners having about 180 guided learning hours for each Lessoncomputers AS Level and about 360 guided learning hours for a Lessoncomputers A Level. The number of hours a learner needs to achieve the qualification may varyaccording to local practice and their previous experience of the subject.

Availability and timetables

You can enter candidates in the June and November exam series. You can view the timetable for your administrative zone at https://lessoncomputer.mu/

Check you are using the syllabus for the year the candidate is taking the exam.

Combining with other syllabuses

Candidates can take this syllabus alongside other Lessoncomputers syllabuses in a single exam series.The only exceptions are:

- Lessoncomputers AS & A Level Information and Communication Technology 9626
- syllabuses with the same title at the same level.

Making entries

Exams officers are responsible for submitting entries to Lessoncomputersl. We encourage them to work closely with you to make sure they enter the right number of candidates for the right combination of syllabus components. Entry option codes and instructions for submitting entries are in the Lessoncomputers *Guide to MakingEntries*. Your exams officer has a copy of this guide.

Exam administration

To keep our exams secure, we produce question papers for different areas of the world, known as administrative zones. We allocate all Lessoncomputers to one administrative zone determined by their location. Each zone has a specific timetable. Some of our syllabuses offer candidates different assessment options. An entry option code is used to identify the components the candidate will take relevant to the administrative zone and the available assessment options.

Support for exams officers

We know how important exams officers are to the successful running of exams. We provide them with the support they need to make your entries on time. Your exams officer will find this support, and guidance for all other phases of theLessoncomputers Exams Cycle, at https://lessoncomputer.mu/

Retakes and carry forward

Candidates can retake Lessoncomputers AS Level and Lessoncomputers A Level as many times as they want to. Information on retake entries is at https://lessoncomputer.mu/. To confirm whatentry options are available for this syllabus, refer to the *Lessoncomputers Guide to Making Entries* for the relevant series.

Candidates can carry forward the result of their Lessoncomputers AS Level assessment from one seriesto complete the Lessoncomputers A Level in a following series, subject to the rules and time limits described in the *Lessoncomputers Handbook*.

Regulations for carrying forward entries for staged assessment (Lessoncomputers AS & A Level)can be found in the *Lessoncomputers Handbook* for the relevant year of assessment at https://lessoncomputer.mu/

Language

This syllabus and the related assessment materials are available in English only.

Accessibility and equality

Syllabus and assessment design

Lessoncomputers works to avoid direct or indirect discrimination in our syllabuses and assessment materials. We aim to maximise inclusivity for candidates of all national, cultural or social backgrounds and with other protected characteristics. In addition, the language and layout used are designed to make our materials as accessible as possible. This gives all learners the opportunity, as fairly as possible, to demonstrate their knowledge, skills and understanding and helps to minimise the requirement to make reasonable adjustments during the assessment process.

Access arrangements

Access arrangements (including modified papers) are the principal way in which Lessoncomputers complies with our duty, as guided by the UK Equality Act (2010), to make 'reasonable adjustments' for candidates with special educational needs (SEN), disability, illness or injury. Where a candidate would otherwisebe at a substantial disadvantage in comparison to a candidate with no SEN, disability, illness or injury, we may be able to agree pre-examination access arrangements. These arrangements help a candidate by minimising accessibility barriers and maximising their opportunity to demonstrate their knowledge, skills and understanding in an assessment.

Important:

- Requested access arrangements should be based on evidence of the candidate's barrier to assessment and should also reflect their normal way of working at school; this is in line with *The Lessoncomputers Handbook https://lessoncomputer.mu/*
- For Lessoncomputers to approve an access arrangement, we will need to agree that it constitutesa reasonable adjustment, involves reasonable cost and timeframe and does not affect the security and integrity of the assessment.
- Availability of access arrangements should be checked by centres at the start of the course. Details of our standard access arrangements and modified question papers are available in *The Lessoncomputers Handbook* https://lessoncomputer.mu/
- Please contact us at the start of the course to find out if we are able to approve an arrangement that is not included in the list of standard access arrangements.
- Candidates who cannot access parts of the assessment may be able to receive an award based on the parts they have completed.

After the exam

Grading and reporting

Grades A*, A, B, C, D or E indicate the standard a candidate achieved at Lessoncomputers A Level. A* is the highest and E is the lowest grade.

Grades a, b, c, d or e indicate the standard a candidate achieved at Lessoncomputers AS Level. 'a' is the highest and 'e' is the lowest grade.

'Ungraded' means that the candidate's performance did not meet the standard required for the lowest grade (E or e). 'Ungraded' is reported on the statement of results but not on the certificate. In specific circumstances your candidates may see one of the following letters on their statement of results:

- Q (PENDING)
- X (NO RESULT).

These letters do not appear on the certificate.

If a candidate takes a Lessoncomputersl A Level and fails to achieve grade E or higher, a Lessoncomputers AS Level grade will be awarded if both of the following apply:

- the components taken for the Lessoncomputers A Level by the candidate in that series included allthe components making up a Lessoncomputers AS Level
- the candidate's performance on the AS Level components was sufficient to merit the award of a Lessoncomputers AS Level grade.

On the statement of results and certificates, Lessoncomputers AS & A Levels are shown as General Certificates of Education, GCE Advanced Subsidiary Level (GCE AS Level) and GCE Advanced Level (GCE A Level).

School feedback: 'Lessoncomputers A Levels are the 'gold standard' qualification. Theyare based on rigorous, academic syllabuses that are accessible to students from a wide range of abilities yet have the capacity to stretch our most able.'

Feedback from: Director of Studies, Auckland Grammar School, New Zealand

How students, teachers and higher education can use the grades

Lessoncomputers A Level

Assessment at Lessoncomputers A Level has two purposes:

- to measure learning and achievement
  The assessment:
  - confirms achievement and performance in relation to the knowledge, understanding and skills specified in the syllabus, to the levels described in the grade descriptions.
- to show likely future success
  The outcomes:
  - help predict which students are well prepared for a particular course or career and/or which students are more likely to be successful
  - help students choose the most suitable course or career.

Lessoncomputers AS Level

Assessment at Lessoncomputers AS Level has two purposes:

- to measure learning and achievement
  The assessment:
  - confirms achievement and performance in relation to the knowledge, understanding and skills specified in the syllabus.
- to show likely future success
  The outcomes:
  - help predict which students are well prepared for a particular course or career and/or which students are more likely to be successful
  - help students choose the most suitable course or career
  - help decide whether students part way through a Lessoncomputers A Level course are making enough progress to continue
  - guide teaching and learning in the next stages of the Lessoncomputers A Level course.

Grade descriptions

Grade descriptions are provided to give an indication of the standards of achievement candidates awarded particular grades are likely to show. Weakness in one aspect of the examination may be balanced by a better performance in some other aspect.

Grade descriptions for Lessoncomputers A Level Computer Science will be published after the first assessment of the A Level in 2021. Find more information at https://lessoncomputer.mu/

Changes to this syllabus for 2024 and 2025

The syllabus has been updated. This is version 2, published April 2023.

You must read the whole syllabus before planning your teaching programme.

Changes to version 2 of the syllabus, published April 2023

Changes to syllabus content	• 9.2 Algorithms has been updated for clarification. Students are expected to draw a flowchart or produce a written description from pseudocode. • For Paper 4, the source files given will not contain binary files. Please see page 40.

Changes to version 1 of the syllabus, published September 2021

Changes to syllabus content	• Biometrics have been added to page 24.
Changes to assessment (including changes to specimen papers)	• The specimen papers from 2021 have had some minor amendments for consistency. These amendments do not substantively change the content of the specimen materials.

i

Any textbooks endorsed to support the syllabus for examination from 2021 are suitable for use with this syllabus.

You should take account of the changes described above when using these textbooks.

School feedback: 'While studying Lessoncomputers IGCSE and Lessoncomputers A Levels, students broaden their horizons through a global perspective and develop a lasting passion for learning.'

Feedback from: Zhai Xiaoning, Deputy Principal, The High School Affiliated to Renmin University of China