edexcel

INTERNATIONAL GCSE

Computer Science (9-1)

SPECIFICATION

Pearson Edexcel International GCSE in Computer Science (4CP0)

For first teaching September 2017
First examination June 2019

Pearson

INTERNATIONAL GCSE

Computer Science

SPECIFICATION

Pearson Edexcel International GCSE in Computer Science (4CP0)

For first teaching in September 2017

First examination June 2019

Edexcel, BTEC and LCCI qualifications

Edexcel, BTEC and LCCI qualifications are awarded by Pearson, the UK's largest awarding body offering academic and vocational qualifications that are globally recognised and benchmarked. For further information, please visit our qualifications website at qualifications.pearson.com. Alternatively, you can get in touch with us using the details on our contact us page at qualifications.pearson.com/contactus

About Pearson

Pearson is the world's leading learning company, with 35,000 employees in more than 70 countries working to help people of all ages to make measurable progress in their lives through learning. We put the learner at the centre of everything we do, because wherever learning flourishes, so do people. Find out more about how we can help you and your learners at qualifications.pearson.com

Acknowledgements

This specification has been produced by Pearson on the basis of consultation with teachers, examiners, consultants and other interested parties. Pearson would like to thank all those who contributed their time and expertise to the specification's development.

ISBN 978 1 446 93428 9

Contents

1 About this specification

The Pearson Edexcel International GCSE in Computer Science is part of a suite of International GCSE qualifications offered by Pearson.

This qualification is not accredited or regulated by any UK regulatory body.

This specification includes the following key features.

Structure: the Pearson Edexcel International GCSE in Computer Science is a linear qualification. Two examinations must be taken in the same series at the end of the course of study.

Content: relevant, engaging, up to date and of equivalent standard to Pearson's regulated GCSE in Computer Science.

Assessment: 100% external assessment consisting of two papers. A written examination and a practical programming examination, both with questions and tasks designed to be accessible to students of all abilities.

Approach: independent learning, problem solving, computational thinking skills, critical thinking skills, real-life examples.

Specification updates

This specification is Issue 1 and is valid for the Pearson Edexcel International GCSE in Computer Science examination from 2019. If there are any significant changes to the specification, Pearson will inform centres. Changes will also be posted on our website.

For more information please visit qualifications.pearson.com

Using this specification

This specification has been designed to give guidance to teachers and encourage effective delivery of the qualification. The following information will help you get the most out of the content and guidance.

Compulsory content: arranged as six topics in *Section 2: Computer Science* content. A summary of sub-topics is included at the start of each topic. As a minimum, all the bullet points in the content must be taught. The word 'including' in content specifies the detail of what must be covered.

Assessments: use a range of material and are not limited to the examples given. Teachers should deliver the qualification using a good range of examples to support the assessment of the content.

Depth and breadth of content: teachers should use the full range of content and all the assessment objectives given in *Section 2: Computer Science content*.

Qualification aims and objectives

The aims and objectives of this qualification are to enable students to:

- understand and apply the fundamental principles and concepts of computer science, including abstraction, decomposition, logic, algorithms and data representation
- analyse problems in computational terms through practical experience of solving such problems, including designing, writing and debugging programs
- think creatively, innovatively, analytically, logically and critically
- understand the components that make up digital systems, and how they communicate with one another and with other systems
- understand the impacts of digital technology to the individual and to wider society
- apply mathematical skills relevant to computer science.

Why choose Pearson Edexcel International GCSE in Computer Science?

We've listened to feedback from all parts of the international school subject community and a representative at Intel, developing changes that will engage students and give them the skills that will support progression to further study of computer science, mathematics, physics and engineering.

We believe ICT and computer science are unique and complementary subjects, serving different purposes and meeting different student needs, and therefore we are offering both subjects as International GCSEs.

Taking an International GCSE in Computer Science will enable students to meet the demands of 21st century employers.

A CBI Report stated that:

> *'teaching programming and computer science in much greater depth will expand understanding of the digital world … encouraging the development of the active, creative interactions with technology that will be fundamental to economic success in the 21st century. Employers need people who are not only effective users of technology, but also able to innovate with it. Part of this goes beyond technical skills to the behaviours and competencies that are in part developed by engaging with the digital creative process'*

Source: CBI response: UK Digital Skills Taskforce – Call for Evidence, May 2014

The content and assessment approach for this qualification has been designed to meet student needs in the following ways.

Develop computational thinking skills – this qualification provides students with the opportunity to operate confidently in today's digital world, enabling students to apply computational thinking in context, across both written and practical examinations.

Provide practical opportunities – students will be encouraged repeatedly to design, implement and test programs that provide solutions to problems. They will apply their skills to produce robust programs and this will help them to progress to further/higher education where practical knowledge and experience will be required.

Range of programming languages – we will provide a choice of three programming languages, allowing flexibility for centres and students to make choices that are the most valuable and appropriate for them.

Clear and straightforward question papers – our question papers are clear and accessible for students of all ability ranges and learning styles. Our mark schemes are straightforward and the assessment requirements are clear.

Broad and deep development of students' skills – the International GCSE is designed to extend students' knowledge and understanding by broadening and deepening skills. For example students will develop the ability to:

- apply the fundamental principles and concepts of computer science, including abstraction, decomposition, logic, algorithms and data representation
- analyse problems in computational terms through practical problem-solving experience. This will include designing, writing and debugging programs
- think creatively, innovatively, analytically, logically and critically
- apply mathematical skills relevant to computer science.

Progression to A Level – International GCSEs enable successful progression to A Level and beyond. Through our world-class qualification development process, we have consulted with teachers, as well as university professors and an Intel representative, to validate the appropriateness of this qualification for progression, including its content, opportunities for skills development and assessment structure.

Centres wishing to teach students how to use computer systems can use our Pearson Edexcel International GCSE in ICT. We offer both Computer Science and ICT, which can be taken in combination, to enable our centres to choose the most appropriate qualification for each student.

More information about all of our qualifications can be found on our Edexcel International GCSE pages at: qualifications.pearson.com

Supporting you in planning and implementing this qualification

Planning

- Our *Getting Started Guide* gives you an overview of the Pearson Edexcel International GCSE in Computer Science to help you understand the content and assessment.
- We will provide you with a course planner and editable schemes of work.

Teaching and learning

- Our skills maps will highlight opportunities for students to develop skills that are assessed as well as skills that are not directly assessed.

Preparing for exams

We will also provide a range of resources to help you prepare your students for the assessments, including:

- specimen papers to support formative assessments and mock exams
- examiner commentaries following each examination series.

ResultsPlus

ResultsPlus provides the most detailed analysis available of your students' exam performance. It can help you identify the topics and skills where further learning would benefit your students.

Training events

In addition to online training, we host a series of training events each year for teachers to deepen their understanding of our qualifications.

Get help and support

Our subject advisor service will ensure you receive help and guidance from us. You can sign up to receive Edexcel newsletters which provide qualification updates and product and service news.

Qualification at a glance

The Pearson Edexcel International GCSE in Computer Science is comprised of two externally-assessed papers, one of which is a practical assessment carried out on a computer system using a programing language of choice.

- Paper 1: Principles of Computer Science
- Paper 2: Application of Computational Thinking

Paper overview

<table>
<tr><td>Paper 1: Principles of Computer Science</td><td>*Paper code 4CP0/01</td></tr>
<tr><td>• Externally assessed
• Availability: June
• First assessment: June 2019</td><td>50% of the total International GCSE</td></tr>
<tr><td colspan="2">Content summary
This paper will primarily assess knowledge and understanding of the basic principles of computer science, including some coverage of how these principles are applied when solving problems that relate to a particular situation.
This paper will assess all topics.
• Understanding of what algorithms are, what they are used for and how they work; ability to interpret, amend and create algorithms.
• Understanding the requirements for writing program code.
• Understanding how to develop program code and constructs, data types, structures, input/output, operators and subprograms.
• Understanding of binary representation, data representation, data storage and compression, and encryption.
• Understanding of components of computer systems; ability to construct truth tables, produce logic statements and read and interpret pseudocode.
• Understanding of computer networks, the internet and the world wide web.
• Awareness of emerging trends in computing technologies, the impact of computing on individuals, society and the environment, including ethical, legal and ownership issues.</td></tr>
<tr><td colspan="2">Assessment
• Assessment is through a 2-hour examination paper set and marked by Pearson.
• The paper consists of multiple-choice, short open-response, open-response and extended open-response answer questions.
• The total number of marks available for the examination paper is 80.
• All questions are mandatory.</td></tr>
</table>

<table>
<tr><td>Paper 2: Application of Computational Thinking</td><td>*Paper code 4CP0/02</td></tr>
<tr><td>Externally assessed
Availability: June
First assessment: June 2019</td><td>50% of the total International GCSE</td></tr>
<tr><td colspan="2">Content summary

This paper will primarily assess the practical application of computational thinking, whereby learners will create, use and adapt existing algorithms to solve problems in a particular situation. This paper will also test students’ knowledge and understanding of the topics.

This paper will draw on:
understanding of what algorithms are, what they are used for and how they work; ability to interpret, amend and create algorithms
developing and testing program code and constructs, data types, structures, input/output, operators and subprograms
connecting and using data sources when developing program code
understanding of binary representation, data representation, data storage and compression, and encryption
ability to construct truth tables, produce logic statements and read and interpret pseudocode.</td></tr>
<tr><td colspan="2">Assessment
Assessment is through a 3-hour practical examination, set and marked by Pearson.
A choice of three programming languages will be available (Python, C# or Java).
The paper consists of multiple-choice, short open-response, open-response, extended-open response answer and task-based questions.
The task-based questions will be carried out using a computer system under supervision. All other questions requiring a written response will be answered in the paper.
The total number of marks available for the examination paper is 80.
All questions are mandatory.
A pseudocode reference document will be available for learners to reference during the assessment.</td></tr>
</table>

* See *Appendix 1* for a description of this code and all the other codes relevant to this qualification.

2 Computer Science content

Course structure

- The Pearson Edexcel International GCSE in Computer Science is comprised of two externally-assessed papers assessing content across six topic areas.
- The Pearson Edexcel International GCSE in Computer Science is a linear qualification. Two examinations must be taken in the same series at the end of the course of study.

Topic 1: Problem solving

Students are expected to develop a set of computational thinking skills that enable them to understand how computer systems work, and design, implement and analyse algorithms for solving problems.

Students should be given repeated opportunities to tackle computational problems of various sorts, including some substantial problem-solving tasks.

Students will be expected to use the pseudocode listed in *Appendix 5: Pseudocode command set* and the flowchart symbols shown in *Appendix 6: Flowchart symbols.*

	Students should:
1.1 Algorithms	**1.1.1** Understand what an algorithm is, what algorithms are used for and be able to interpret algorithms (flowcharts, pseudocode, written descriptions, program code). **1.1.2** Understand how to create an algorithm to solve a particular problem, making use of programming constructs (sequence, selection, iteration) and using appropriate conventions (flowchart, pseudocode, written description, draft program code). **1.1.3** Understand the purpose of a given algorithm and how an algorithm works. **1.1.4** Understand how to determine the correct output of an algorithm for a given set of data. **1.1.5** Understand how to identify and correct errors in algorithms, including using trace tables. **1.1.6** Understand how to code an algorithm in a high-level language. **1.1.7** Understand how the choice of algorithm is influenced by the data structures and data values that need to be manipulated. **1.1.8** Understand how standard algorithms work (bubble sort, merge sort, linear search, binary search). **1.1.9** Be able to evaluate the fitness for purpose of algorithms in meeting specified requirements efficiently, using logical reasoning and test data.

	Students should:
1.2 Decomposition and abstraction	**1.2.1** Be able to analyse a problem, investigate requirements (inputs, outputs, processing, initialisation) and design solutions. **1.2.2** Be able to decompose a problem into smaller sub-problems. **1.2.3** Understand how abstraction can be used effectively to model aspects of the real world. **1.2.4** Be able to program abstractions of real-world examples.

Topic 2: Programming

Learning to program is a core component of a computer science course. Students should be competent at designing, reading, writing and debugging programs. They must be able to apply their skills to solve real problems and produce robust programs.

Students should be given repeated opportunities to develop and practise their programming skills. They will develop their knowledge and skills in at least one of the following programming languages:

1. Python
2. C#
3. Java.

	Students should:
2.1 Develop code	**2.1.1** Be able to write programs in a high-level programming language. **2.1.2** Understand the benefit of producing programs that are easy to read and be able to use techniques (comments, descriptive names (variables, constants, subprograms), indentation) to improve readability and to explain how the code works. **2.1.3** Be able to differentiate between types of error in programs (logic, syntax, runtime). **2.1.4** Be able to design and use test plans and test data (normal, boundary, erroneous). **2.1.5** Be able to interpret error messages and identify, locate and fix errors in a program. **2.1.6** Be able to determine what value a variable will hold at a given point in a program (trace table). **2.1.7** Be able to determine the strengths and weaknesses of a program and suggest improvements.
2.2 Constructs	**2.2.1** Understand the structural components of a program (variable and type declarations, command sequences, selection, iteration, data structures, subprograms). **2.2.2** Be able to use sequencing, selection and iteration constructs in their programs.

	Students should:
2.3 Data types and structures	**2.3.1** Understand the need for, and understand how to use, data types (integer, real, Boolean, char, string). **2.3.2** Understand the need for, and understand how to use, data structures (records, one-dimensional arrays, two-dimensional arrays). **2.3.3** Understand the need for, and how to manipulate, strings. **2.3.4** Understand the need for, and how to use, variables and constants. **2.3.5** Understand the need for, and how to use, global and local variables when implementing subprograms.
2.4 Input/output	**2.4.1** Understand how to write code that accepts and responds appropriately to user input. **2.4.2** Understand the need for, and how to implement, validation. **2.4.3** Be able to write code that reads/writes from/to a text file.
2.5 Operators	**2.5.1** Understand the purpose of, and how to use, arithmetic operators (add, subtract, divide, multiply, modulus, integer division). **2.5.2** Understand the purpose of, and how to use, relational operators (equal to, less than, greater than, not equal to, less than or equal to, greater than or equal to). **2.5.3** Understand the purpose of, and how to use, logic operators (AND, OR, NOT).
2.6 Subprograms	**2.6.1** Understand the benefits of using subprograms and be able to write code that uses user-written and pre-existing (built-in, library) subprograms. **2.6.2** Understand the concept of passing data into and out of subprograms (procedures, functions). **2.6.3** Be able to create subprograms that use parameters.

Topic 3: Data

Computers are able to store and manipulate large quantities of data. They use binary to represent different types of data.

Students are expected to learn how different types of data are represented in a computer.

	Students should:
3.1 Binary	**3.1.1** Understand that computers use binary to represent data (numbers, text, sound, graphics) and program instructions. **3.1.2** Understand how computers represent and manipulate numbers (unsigned integers, signed integers (sign and magnitude, two's complement)). **3.1.3** Be able to convert between binary and denary whole numbers (0–255). **3.1.4** Understand how to perform binary arithmetic (add, shifts (logical and arithmetic)) and understand the concept of overflow. **3.1.5** Understand why hexadecimal notation is used and be able to convert between hexadecimal and binary.
3.2 Data representation	**3.2.1** Understand how computers encode characters using ASCII and Unicode. **3.2.2** Understand how bitmap images are represented in binary (pixels, resolution, colour depth). **3.2.3** Understand how sound, an analogue signal, is represented in binary. **3.2.4** Understand the limitations of binary representation of data (sampling frequency, resolution) when constructed by the number of available bits.

	Students should:
3.3 Data storage and compression	**3.3.1** Understand how to use and convert between binary and denary multiples (as defined by the International Electrotechnical Commission (IEC)): • bit • nibble • byte • kibibyte (KiB) 2^{10} • mebibyte (MiB) 2^{20} • gibibyte (GiB) 2^{30} • tebibyte (TiB) 2^{40} • kilobyte (kB), 10^{3} • megabyte (MB) 10^{6} • gigabyte (GB) 10^{9} • terabyte (TB) 10^{12}. **3.3.2** Understand the need for data compression and methods of compressing data (lossless, lossy), and that JPEG and MP3 are examples of lossy algorithms. **3.3.3** Understand how a lossless, run-length encoding (RLE) algorithm works. **3.3.4** Understand that file storage is measured in bytes and be able to calculate file sizes.
3.4 Encryption	**3.4.1** Understand the need for data encryption. **3.4.2** Understand how encryption algorithms work (Pigpen cipher, Caesar cipher, Vigenére cipher, Rail Fence cipher).

Topic 4: Computers

Students must be familiar with the hardware and software components that make up a computer system and recognise that computers take many forms, from embedded microprocessors to distributed clouds.

	Students should:
4.1 Machines and computational modelling	**4.1.1** Understand the input-process-output model. **4.1.2** Understand that there is a range of computational models (sequential, parallel, multi-agent).
4.2 Hardware	**4.2.1** Understand the function of the hardware components of a computer system (central processing unit (CPU), main memory, secondary storage, input and output devices) and how they work together. **4.2.2** Understand the function of different types of memory (random-access memory (RAM), read-only memory (ROM), cache, virtual memory). **4.2.3** Understand the concept of a stored program and the role of components of the CPU (control unit (CU), arithmetic/logic unit (ALU), registers, clock, address bus, data bus, control bus) in the fetch-decode-execute cycle (the Von Neumann model). **4.2.4** Understand factors that affect the performance of the CPU (clock speed, number of processor cores, size of cache, type of cache). **4.2.5** Understand how data is stored on physical devices (magnetic, optical, solid state). **4.2.6** Understand the concept of storing data in the 'cloud' and other contemporary secondary storage. **4.2.7** Understand the need for embedded systems and their functions.
4.3 Logic	**4.3.1** Be able to construct and interpret truth tables for a given logic statement (AND, OR, NOT). **4.3.2** Be able to produce logic statements for a given problem.

	Students should:
4.4 Software	**4.4.1** Know what an operating system is and how it manages files, processes, hardware and the user interface. **4.4.2** Understand the purpose and functions of utility software (managing, repairing and converting files; compression; defragmentation; backing up; anti-malware (antivirus, anti-spyware)). **4.4.3** Understand how software can be used to simulate and model aspects of the real world. **4.4.4** Understand what is meant by system software and application software.
4.5 Programming languages	**4.5.1** Understand what is meant by high-level and low-level programming languages and understand their suitability for a particular task. **4.5.2** Understand what is meant by an assembler, a compiler and an interpreter when translating programming languages and know the advantages and disadvantages of each.

Topic 5: Communication and the internet

Computer networks and the internet are now ubiquitous. Many computer applications in use today would not be possible without networks. Students should understand the key principles behind the organisation of computer networks. Ideally, they should be able to experiment by setting up a simple network. Our increasing reliance on computer networks makes us vulnerable to a range of attacks from cyber criminals. As IT system security defences become more robust, attack methods become more sophisticated. Students should have a good understanding of the current IT security threats and how to apply appropriate protection methods.

	Students should:
5.1 Networks	**5.1.1** Understand why computers are connected in a network. **5.1.2** Understand the different types of networks (local area network (LAN), wide area network (WAN), personal area network (PAN)) and usage models (client-server, peer-to-peer). **5.1.3** Understand wired and wireless connectivity. **5.1.4** Understand that network data speeds are measured in bits per second (Mbps, Gbps). **5.1.5** Understand the role of and need for network protocols (Ethernet®, Wi-Fi, TCP/IP, HTTP, HTTPS, FTP, email (POP3, SMTP, IMAP)). **5.1.6** Understand that data can be transmitted in packets using layered protocol stacks and the 4-layer TCP/IP model (application, transport, network, data link). **5.1.7** Understand characteristics of network topologies (bus, ring, star, mesh). **5.1.8** Understand the different mobile communication standards (3G, 4G and subsequent generations).

	Students should:
5.2 Network security	**5.2.1** Understand the importance of network security and be able to use appropriate validation and authentication techniques (access control, physical security and firewalls). **5.2.2** Understand security issues associated with the 'cloud' and other contemporary storage. **5.2.3** Understand different forms of cyber attack (based on technical weaknesses and behaviour), including social engineering (phishing, shoulder surfing, pharming), unpatched software, USB devices, digital devices and eavesdropping. **5.2.4** Understand methods of identifying vulnerabilities, including penetration testing, ethical hacking, commercial analysis tools and review of network and user policies. **5.2.5** Understand how to protect software systems from cyber attacks, including considerations at the software (application) design stage, audit trails, securing operating systems, secure coding, code reviews to remove code vulnerabilities in programming languages and bad programming practices, modular testing and effective network security provision.
5.3 The internet and the world wide web	**5.3.1** Understand what is meant by the internet and how the internet is structured (IP addressing, domain name service (DNS)). **5.3.2** Understand what is meant by the world wide web (WWW) and components of the WWW (web server URLs, ISP, HTTP, HTTPS, HTML). **5.3.3** Understand the need for IP addressing standards and the formats of IPv4 and IPv6. **5.3.4** Understand the role of components used to access the internet (modem, router, switch, wireless access point (WAP)) and how these are combined.

Topic 6: The bigger picture

Students should be aware of the influence of computing technology and recognise that computing has an impact on nearly every aspect of the world in which they live.

	Students should:
6.1 Emerging trends, issues and impact	**6.1.1** Understand the environmental impact of technology (health, energy use, resources) on society. **6.1.2** Understand the ethical impact of using technology (privacy, inclusion, professionalism) on society. **6.1.3** Understand the legal impact of using technology (intellectual property, patents, licensing and cyber-security). **6.1.4** Be aware of current and emerging trends in computing technology (quantum computing, DNA computing, artificial intelligence (AI), nanotechnology).

3 Assessment information

Assessment requirements

Paper number and unit title	Level	Assessment information	Number of raw marks allocated in the paper
Paper 1: Principles of Computer Science	1/2	Assessed through a 2-hour written examination set and marked by Pearson. The paper is weighted at 50% of the qualification. The paper consists of multiple-choice, short open-response, open-response and extended open-response answer questions.	80
Paper 2: Application of Computational Thinking	1/2	Assessed through a 3-hour practical examination set and marked by Pearson. The paper is weighted at 50% of the qualification.	80

Assessment objectives and weightings

		% in International GCSE
AO1	Demonstrate knowledge and understanding of the key principles of computer science	27.5
AO2	Apply knowledge and understanding of key concepts and principles of computer science	42.5
AO3	Analyse problems in computational terms: • to make reasoned judgements • to design, program, test, evaluate and refine solutions	30.0

Relationship of assessment objectives to units

Unit number	**Assessment objective**		
	AO1	**AO2**	**AO3**
Paper 1: Principles of Computer Science	21.5%	21%	7.5%
Paper 2: Application of Computational Thinking	6%	21.5%	22.5%
Total for International GCSE	27.5%	42.5%	30%

All papers will be available for assessment from June 2019.

4 Administration and general information

Entries

Details of how to enter students for the examinations for this qualification can be found in our *International Information Manual*. A copy is made available to all examinations officers and is available on our website.

Students should be advised that if they take two qualifications in the same subject, colleges, universities and employers are very likely to take the view that they have achieved only one of the two GCSEs/International GCSEs. Students or their advisers who have any doubts about subject combinations should check with the institution to which they wish to progress before embarking on their programmes.

Access arrangements, reasonable adjustments, special consideration and malpractice

Equality and fairness are central to our work. Our equality policy requires all students to have equal opportunity to access our qualifications and assessments, and our qualifications to be awarded in a way that is fair to every student.

We are committed to making sure that:

- students with a protected characteristic (as defined by the UK Equality Act 2010) are not, when they are undertaking one of our qualifications, disadvantaged in comparison to students who do not share that characteristic
- all students achieve the recognition they deserve for undertaking a qualification and that this achievement can be compared fairly to the achievement of their peers.

Language of assessment

Assessment of this qualification will only be available in English. All student work must be in English.

We recommend that students are able to read and write in English at Level B2 of the Common European Framework of Reference for Languages.

Arrangements for Paper 2

Assessment is available within a window. The following requirements must be adhered to.

- Centres must ensure that there are suitable IT facilities available for learners to be assessed.
- The dates of the window will be published by Pearson.
- Paper sittings must be scheduled to prevent the possibility of learners colluding.
- Learners must only have access to files required for the paper.
- During the test, learners are **not** permitted internet access.
- Teaching of the subject should be suspended for the duration of the window.

Full details will be made available in the *Instructions for the Conduct of the Examination (ICE)* document published on the Pearson website.

Access arrangements

Access arrangements are agreed before an assessment. They allow students with special educational needs, disabilities or temporary injuries to:

- access the assessment
- show what they know and can do without changing the demands of the assessment.

The intention behind an access arrangement is to meet the particular needs of an individual student with a disability without affecting the integrity of the assessment. Access arrangements are the principal way in which awarding bodies comply with the duty under the Equality Act 2010 to make 'reasonable adjustments'.

Access arrangements should always be processed at the start of the course. Students will then know what is available and have the access arrangement(s) in place for assessment.

Reasonable adjustments

The Equality Act 2010 requires an awarding organisation to make reasonable adjustments where a student with a disability would be at a substantial disadvantage in undertaking an assessment. The awarding organisation is required to take reasonable steps to overcome that disadvantage.

A reasonable adjustment for a particular student may be unique to that individual and therefore might not be in the list of available access arrangements.

Whether an adjustment will be considered reasonable will depend on a number of factors, including:

- the needs of the student with the disability
- the effectiveness of the adjustment
- the cost of the adjustment; and
- the likely impact of the adjustment on the student with the disability and other students.

An adjustment will not be approved if it involves unreasonable costs to the awarding organisation, timeframes or affects the security or integrity of the assessment. This is because the adjustment is not 'reasonable'.

Special consideration

Special consideration is a post-examination adjustment to a student's mark or grade to reflect temporary injury, illness or other indisposition at the time of the examination/ assessment, which has had, or is reasonably likely to have had, a material effect on a candidate's ability to take an assessment or demonstrate his or her level of attainment in an assessment.

Further information

Please see our website for further information about how to apply for access arrangements and special consideration.

For further information about access arrangements, reasonable adjustments and special consideration please refer to the JCQ website: www.jcq.org.uk.

Candidate malpractice

Candidate malpractice refers to any act by a candidate that compromises or seeks to compromise the process of assessment or which undermines the integrity of the qualifications or the validity of results/certificates.

Candidate malpractice in examinations **must** be reported to Pearson using a *JCQ Form M1* (available at www.jcq.org.uk/exams-office/malpractice). The form can be emailed to pqsmalpractice@pearson.com or posted to: Investigations Team, Pearson, 190 High Holborn, London, WC1V 7BH. Please provide as much information and supporting documentation as possible. Note that the final decision regarding appropriate sanctions lies with Pearson.

Failure to report malpractice constitutes staff or centre malpractice.

Staff/centre malpractice

Staff and centre malpractice includes both deliberate malpractice and maladministration of our qualifications. As with candidate malpractice, staff and centre malpractice is any act that compromises or seeks to compromise the process of assessment or which undermines the integrity of the qualifications or the validity of results/certificates. The security of assessment materials, including secure data files, must be maintained at all times prior to and throughout the examination window. Failure to do so may constitute maladministration or malpractice. Any breach must be reported to Pearson immediately at pqsmalpractice@pearson.com.

All cases of suspected staff malpractice and maladministration **must** be reported immediately, before any investigation is undertaken by the centre, to Pearson on a *JCQ Form M2(a)* (available at www.jcq.org.uk/exams-office/malpractice).

The form, supporting documentation and as much information as possible can be emailed to pqsmalpractice@pearson.com or posted to: Investigations Team, Pearson, 190 High Holborn, London, WC1V 7BH. Note that the final decision regarding appropriate sanctions lies with Pearson.

Failure to report malpractice itself constitutes malpractice.

More-detailed guidance on malpractice can be found in the latest version of the document *JCQ General and vocational qualifications Suspected Malpractice in Examinations and Assessments,* available at www.jcq.org.uk/exams-office/malpractice.

Awarding and reporting

The International GCSE qualification will be graded and certificated on a nine-grade scale from 9 to 1 using the total subject mark where 9 is the highest grade. Individual components are not graded. The first certification opportunity for the Pearson Edexcel International GCSE in Computer Science will be in 2019. Students whose level of achievement is below the minimum judged by Pearson to be of sufficient standard to be recorded on a certificate will receive an unclassified U result.

Student recruitment and progression

Pearson’s policy concerning recruitment to our qualifications is that:

- they must be available to anyone who is capable of reaching the required standard
- they must be free from barriers that restrict access and progression
- equal opportunities exist for all students.

Prior learning and other requirements

There are no prior learning or other requirements for this qualification.

Progression

The Pearson Edexcel International GCSE in Computer Science develops students’ knowledge, understanding and skills in computing as a basis for progression to:

- International Advanced Level in Applied ICT
- other equivalent Level 3 qualifications
- vocational qualifications, such as BTEC Nationals.

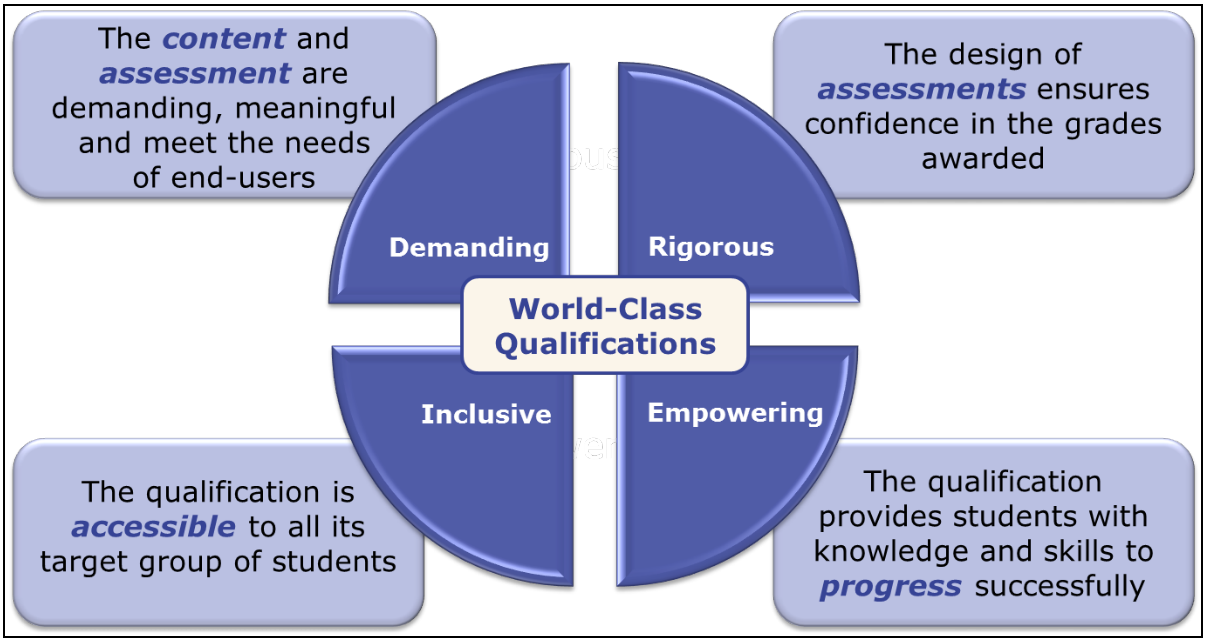
Appendices

Appendix 1: Codes

Type of code	Use of code	Code
Subject codes	The subject code is used by centres to cash in the entry for a qualification.	4CP0
Paper codes	These codes are provided for information. Students may need to be entered for individual papers.	Paper 1: 4CP0/01 Paper 2: 4CP0/02

Appendix 2: Pearson World-Class Qualification Design Principles

Pearson's world-class qualification design principles mean that all Pearson qualifications are developed to be **rigorous, demanding, inclusive and empowering**.



We work collaboratively to gain approval from an external panel of educational thought-leaders and assessment experts from across the globe. This is to ensure that Pearson qualifications are globally relevant, represent world-class best practice in qualification and assessment design, maintain a consistent standard and support student progression in today's fast-changing world.

Pearson's Expert Panel for World-class Qualifications is chaired by Sir Michael Barber, a leading authority on education systems and reform. He is joined by a wide range of key influencers with expertise in education and employability.

"*I'm excited to be in a position to work with the global leaders in curriculum and assessment to take a fresh look at what young people need to know and be able to do in the 21st century, and to consider how we can give them the opportunity to access that sort of education.*" Sir Michael Barber.

Endorsement from Pearson's Expert Panel for World-class Qualifications for International GCSE development processes

"We were chosen, either because of our expertise in the UK education system, or because of our experience in reforming qualifications in other systems around the world as diverse as Singapore, Hong Kong, Australia and a number of countries across Europe.

We have guided Pearson through what we judge to be a rigorous world-class qualification development process that has included:

- Extensive international comparability of subject content against the highest-performing jurisdictions in the world
- Benchmarking assessments against UK and overseas providers to ensure that they are at the right level of demand
- Establishing External Subject Advisory Groups, drawing on independent subject-specific expertise to challenge and validate our qualifications

Importantly, we have worked to ensure that the content and learning is future oriented, and that the design has been guided by Pearson's Efficacy Framework. This is a structured, evidence-based process which means that student outcomes have been at the heart of this development throughout.

We understand that ultimately it is excellent teaching that is the key factor to a learner's success in education but as a result of our work as a panel we are confident that we have supported the development of Pearson Edexcel International GCSE qualifications that are outstanding for their coherence, thoroughness and attention to detail and can be regarded as representing world-class best practice."

Sir Michael Barber (Chair)

Chief Education Advisor, Pearson plc

Professor Sing Kong Lee

Professor, National Institute of Education in Singapore

Dr Peter Hill

Former Chief Executive ACARA

Bahram Bekhradnia

President, Higher Education Policy Institute

Professor Jonathan Osborne

Stanford University

Dame Sally Coates

Director of Academies (South), United Learning Trust

Professor Dr Ursula Renold

Federal Institute of Technology, Switzerland

Professor Bob Schwartz

Harvard Graduate School of Education

Professor Janice Kay

Provost, University of Exeter

Jane Beine

Head of Partner Development, John Lewis Partnership

Jason Holt

CEO, Holts Group

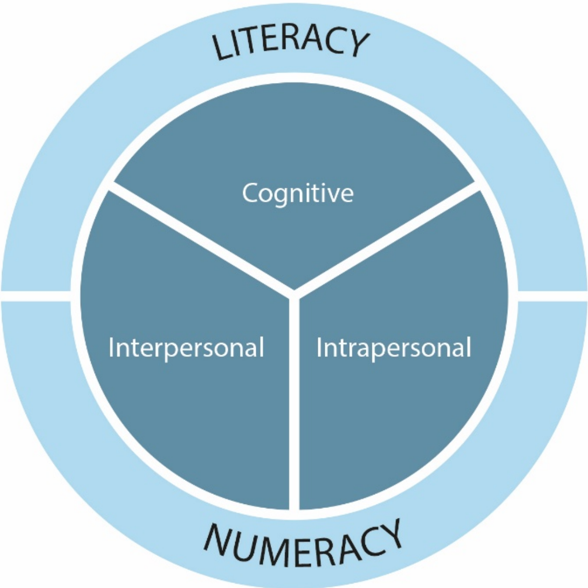
Appendix 3: Transferable skills

The need for transferable skills

In recent years, higher education institutions and employers have consistently flagged the need for students to develop a range of transferable skills to enable them to respond with confidence to the demands of undergraduate study and the world of work.

The Organisation for Economic Co-operation and Development (OECD) defines skills, or competencies, as 'the bundle of knowledge, attributes and capacities that can be learned and that enable individuals to successfully and consistently perform an activity or task and can be built upon and extended through learning.'[1]

To support the design of our qualifications, the Pearson Research Team selected and evaluated seven global 21st-century skills frameworks. Following on from this process, we identified the National Research Council's (NRC) framework [2] as the most evidence-based and robust skills framework, and have used this as a basis for our adapted skills framework.



The skills have been interpreted for this specification to ensure they are appropriate for the subject. All of the skills listed are evident or accessible in the teaching, learning and/or assessment of the qualification. Some skills are directly assessed. Pearson materials will support you in identifying these skills and developing these skills in students.

The table overleaf sets out the framework and gives an indication of the skills that can be found in Computer Science and indicates the interpretation of the skill in this area. A full subject interpretation of each skill, with mapping to show opportunities for students' development, is provided on the subject pages of our website: qualifications.pearson.com

---

[1] OECD – *Better Skills, Better Jobs, Better Lives: A Strategic Approach to Skills Policies* (OECD Publishing, 2012) http://skills.oecd.org/documents/OECDSkillsStrategyFINALENG.pdf

[2] Koenig J A – *Assessing 21st Century Skills: Summary of a Workshop* (National Research Council, 2011)

Cognitive skills	Cognitive processes and strategies	• Critical thinking • Problem solving • Analysis • Reasoning • Interpretation • Decision making • Adaptive learning • Executive function
	Creativity	• Creativity • Innovation
Intrapersonal skills	Intellectual openness	• Adaptability • Personal and social responsibility • Continuous learning • Intellectual interest and curiosity
	Work ethic/ conscientiousness	• Initiative • Self-direction • Responsibility • Perseverance • Productivity • Self-regulation (metacognition, forethought, reflection) • Ethics • Integrity
	Positive core self-evaluation	• Self-monitoring/ self-evaluation/ self-reinforcement
Interpersonal skills	Teamwork and collaboration	• Communication • Collaboration • Teamwork • Cooperation • Interpersonal skills
	Leadership	• Leadership • Responsibility • Assertive communication • Self-presentation

Problem solving for Computer Science

Using computational thinking skills to develop code

Perseverance for Computer Science

Using program testing tools and techniques to remove errors in code

Communication for Computer Science

Communicate ideas to peers and discuss the logic of coding

Appendix 4: Glossary

Term	Definition
Assessment objectives	The requirements that students need to meet to succeed in the qualification. Each assessment objective has a unique focus which is then targeted in examinations or coursework. Assessment objectives may be assessed individually or in combination.
External assessment	An examination that is held at the same time and place in a global region.
JCQ	Joint Council for Qualifications. This is a group of UK exam boards who develop policy related to the administration of examinations.
Linear	Qualifications that are linear have all assessments at the end of a course of study. It is not possible to take one assessment earlier in the course of study.
Raw marks	Raw marks are the actual marks that students achieve when taking an assessment. When calculating an overall grade raw marks often need to be converted so that it is possible to see the proportionate achievement of a student across all units of study.

Appendix 5: Pseudocode command set

Questions in the written examination that involve code will use this pseudocode for clarity and consistency. However, students may answer questions using any valid method.

Data types

INTEGER

REAL

BOOLEAN

CHARACTER

Type coercion

Type coercion is automatic if indicated by context. For example 3 + 8.25 = 11.25 (integer + real = real)

Mixed mode arithmetic is coerced like this:

	INTEGER	REAL
INTEGER	INTEGER	REAL
REAL	REAL	REAL

Coercion can be made explicit. For example, RECEIVE age FROM (INTEGER) KEYBOARD assumes that the input from the keyboard is interpreted as an INTEGER, not a STRING.

Constants

The value of constants can only ever be set once. They are identified by the keyword CONST. Two examples of using a constant are shown.

CONST REAL PI

SET PI TO 3.14159

SET circumference TO radius * PI * 2

Data structures

ARRAY

STRING

Indices start at zero (0) for all data structures.

All data structures have an append operator, indicated by &.

Using & with a STRING and a non-STRING will coerce to STRING. For example, SEND 'Fred' & age TO DISPLAY, will display a single STRING of 'Fred18'.

Identifiers

Identifiers are sequences of letters, digits and '_', starting with a letter, for example: MyValue, myValue, My_Value, Counter2

Functions

LENGTH()

For data structures consisting of an array or string.

RANDOM(n)

This generates a random number from 0 to n.

Comments

Comments are indicated by the # symbol, followed by any text.

A comment can be on a line by itself or at the end of a line.

Devices

Use of KEYBOARD and DISPLAY are suitable for input and output.

Additional devices may be required, but their function will be obvious from the context. For example, CARD_READER and MOTOR are two such devices.

Notes

In the following pseudocode, the < > indicates where expressions or values need to be supplied. The < > symbols are not part of the pseudocode.

Variables and arrays

Syntax	Explanation of syntax	Example
SET Variable TO <value>	Assigns a value to a variable.	SET Counter TO 0 SET MyString TO 'Hello world'
SET Variable TO <expression>	Computes the value of an expression and assigns to a variable.	SET Sum TO Score + 10 SET Size to LENGTH(Word)
SET Array[index] TO <value>	Assigns a value to an element of a one-dimensional array.	SET ArrayClass[1] TO 'Ann' SET ArrayMarks[3]TO 56
SET Array TO [<value>, …]	Initialises a one-dimensional array with a set of values.	SET ArrayValues TO [1, 2, 3, 4, 5]
SET Array [RowIndex, ColumnIndex] TO <value>	Assigns a value to an element of a two-dimensional array.	SET ArrayClassMarks[2,4] TO 92

Selection

Syntax	Explanation of syntax	Example

IF <expression> THEN
   <command>
END IF
	If <expression> is true then command is executed.	IF Answer = 10 THEN
		   SET Score TO Score + 1
		END IF

IF <expression> THEN
   <command>
ELSE
   <command>
END IF
	If <expression> is true then first <command> is executed, otherwise second <command> is executed.	IF Answer = 'correct' THEN
		   SEND 'Well done' TO DISPLAY
		ELSE
		   SEND 'Try again' TO DISPLAY
		END IF

Repetition

Syntax	Explanation of syntax	Example

WHILE <condition> DO
    <command>
END WHILE
	Pre-conditioned loop. Executes
	
	<command> whilst <condition> is true.
		WHILE Flag = 0 DO
		    SEND ‘All well’ TO DISPLAY
		END WHILE

REPEAT <command> UNTIL <expression>	Post-conditioned loop. Executes <command> until <condition> is true. The loop must execute at least once.	REPEAT SET Go TO Go + 1 UNTIL Go = 10
REPEAT <expression> TIMES <command> END REPEAT	Count controlled loop. The number of times <command> is executed is determined by the expression.	REPEAT 100-Number TIMES SEND ‘*’ TO DISPLAY END REPEAT

FOR <id> FROM <expression> TO
<expression> DO
    <command>
END FOR
	Count controlled loop. Executes
	
	<command> a fixed number of times.
		FOR Index FROM 1 TO 10 DO
		    SEND ArrayNumbers[Index] TO DISPLAY
		END FOR

FOR <id> FROM <expression> TO
<expression> STEP
<expression> DO
    <command>
END FOR
	Count controlled loop using a step.	FOR Index FROM 1 TO 500 STEP 25 DO
		    SEND Index TO DISPLAY
		END FOR

FOR EACH <id> FROM <expression> DO <command> END FOREACH	Count controlled loop. Executes for each element of an array.	SET WordsArray TO [‘The’, ‘Sky’, ‘is’, ‘grey’] SET Sentence to ‘‘ FOR EACH Word FROM WordsUArray DO SET Sentence TO Sentence & Word & ‘ ‘ END FOREACH

Input/output		
Syntax	**Explanation of syntax**	**Example**
SEND <expression> TO DISPLAY	Sends output to the screen.	SEND ‘Have a good day.’ TO DISPLAY
RECEIVE <identifier> FROM (type) <device>	Reads input of specified type.	RECEIVE Name FROM (STRING) KEYBOARD RECEIVE LengthOfJourney FROM (INTEGER) CARD_READER RECEIVE YesNo FROM (CHARACTER) CARD_READER

File handling		
Syntax	**Explanation of syntax**	**Example**
READ <File> <record>	Reads in a record from a <file> and assigns to a <variable>. Each READ statement reads a record from the file.	READ MyFile.doc Record
WRITE <File> <record>	Writes a record to a file. Each WRITE statement writes a record to the file.	WRITE MyFile.doc Answer1, Answer2, ‘xyz 01’

Subprograms		
Syntax	**Explanation of syntax**	**Example**
PROCEDURE <id> (<parameter>, …) BEGIN PROCEDURE <command> END PROCEDURE	Defines a procedure.	PROCEDURE CalculateAverage (Mark1, Mark2, Mark3) BEGIN PROCEDURE SET Avg to (Mark1 + Mark2 + Mark3)/3 END PROCEDURE

Subprograms		
Syntax	**Explanation of syntax**	**Example**

FUNCTION <id> (<parameter>, …)
BEGIN FUNCTION
    <command>
RETURN <expression>
END FUNCTION
	Defines a function.	FUNCTION AddMarks (Mark1, Mark2, Mark3)
		BEGIN FUNCTION
		    SET Total to (Mark1 + Mark2 + Mark3)/3
		    RETURN Total
		END FUNCTION

<id> (<parameter>, …)	Calls a procedure or a function.	Add (FirstMark, SecondMark)

Arithmetic operators	
Symbol	**Description**
+	Add
-	Subtract
/	Divide
*	Multiply
^	Exponent
MOD	Modulo
DIV	Integer division

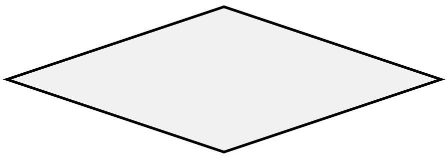
Relational operators	
Symbol	**Description**
=	equal to
<>	not equal to
>	greater than
>=	greater than or equal to
<	less than
<=	less than or equal to

Logical operators	
Symbol	**Description**
AND	Returns true if both conditions are true.
OR	Returns true if any of the conditions are true.
NOT	Reverses the outcome of the expression; true becomes false, false becomes true.

Appendix 6: Flowchart symbols

Denotes the start and end of an algorithm.

Denotes a process to be carried out.



Denotes a decision to be made.

Shows the logical flow of the program.

Denotes a subprocess that may be defined in a separate flowchart.

Denotes input and/or output.

Appendix 7: Taxonomy

The following table lists the command words used in the external assessments.

Command word	Definition
Amend	Requires changes/additions/deletions/rearrangement of a symbolic representation.
Analyse	Examine elements in detail.
Assess	Give careful consideration to all the factors or events that apply and identify which are the most important or relevant. Make a judgement on the importance of something and come to a conclusion where needed.
Calculate	Obtain a numerical answer, showing relevant working. If the answer has a unit, this must be included.
Compare and/or contrast	Looking for the similarities and/or differences of two (or more) things. Should not require the drawing of a conclusion. Answer must relate to both (or all) things mentioned in the question.
Complete	Requires the completion of a table/diagram/algorithm (in any notation).
Construct	Usually requires creation of an artefact using subject specific symbolic representation, rules and syntax.
Convert	Requires changing information in one symbolic representation to another representation. May require amending to provide new functionality/facility.
Create	Usually requires creation of an artefact using subject specific symbolic representations.
Deduce	Draw/reach conclusion(s) from the information provided.
Describe	Give an account of something. Statements in the response need to be developed as they are often linked, but do not need to include a justification or reason.
Devise	Plan or invent a procedure from existing principles/ideas.
Discuss	Identify the issue/situation/problem/argument that is being assessed in the question. Explore all aspects of an issue/situation/problem/argument. Investigate the issue/situation etc. by reasoning or argument.
Draw	Produce a diagram/image, either using a ruler or freehand. May require labelling/annotation to express meaning. Used when symbolic representations need to be manipulated.

Command word	Definition
Evaluate	Review information then bring it together to form a conclusion, drawing on evidence, including strengths, weaknesses, alternative actions, relevant data or information. Come to a supported judgement of a subject's qualities and relation to its context.
Explain	An explanation requires a justification/exemplification of a point. The answer must contain some element of reasoning/justification. This can include mathematical/logical explanations. The mark scheme must have marking points that are linked. The answer MUST contain some element of reasoning/justification to satisfy the definition of explain. The mark scheme should be clearly laid out so that to gain full marks, there must be a minimum of 1 mark for some element of reasoning/justification to satisfy the definition of explain.
Give	Recall one or more pieces of information.
Give a reason/reasons	When a statement has been made and the requirement is only to give the reasons why.
Identify	Usually requires some key information to be selected from a given stimulus/resource.
Name	Recall one piece of information from a given diagram/algorithm/process.
Perform	Carry out an identified task/algorithm/process.
Predict	Give an expected result.
State	Recall one or more pieces of technical information.
State and justify/ Identify and justify	When a selection is made and a justification has to be given for the selection.
State what is meant by	When the meaning of a term is expected but there are different ways to describe them. Usually 1 or 2 marks maximum.
Write	Usually requires manipulation/creation/amending of an artefact using subject specific symbolic representations.

For information about Edexcel, BTEC or LCCI qualifications
visit qualifications.pearson.com

Pearson Education Limited. Registered in England and Wales No. 872828
Registered Office: 80 Strand, London WC2R 0RL
VAT Reg No GB 278 537121

Getty Images: Alex Belmonlinsky

ISBN 978-1-4469-3428-9