

PEARSON EDEXCEL INTERNATIONAL GCSE (9-1) **Computer Science**

Welcome to Pearson

Event code: 4CP0-20IO1

First teaching in 2017, first assessment in 2019.



Welcome to Pearson Edexcel

Welcome to Pearson Edexcel,
the world's leading learning company
and the UK's largest awarding body.

We set the standard for worldwide
recognised qualifications, built on the
UK educational system and accepted
by universities worldwide.

We have a simple mission:
**to help make a measurable impact on
improving people's lives through learning.**

**‘We judge ourselves – and
invite others to judge us –
not by the products that we
make but by the impact on
learners.’**

John Fallon,
Chief Executive Officer,
Pearson



About Pearson Edexcel

- As the UK's largest awarding organisation, we are best placed to provide qualifications that are most closely aligned to the British educational system.
- We are the most reliable awarding organisation in the UK, recognised and trusted by educators, learners and employers to provide high quality qualifications.
- By helping you to realise student potential, you can prepare and empower all your students to progress to further education, university and employment.
- Our technology capability allows us to provide you with more advanced support services, tools and resources to make life easier for school leaders, teachers and students.
- Pearson Edexcel are leading the way, challenging thinking and creating new ideas so you can be confident our qualifications will always be world-class.



Aims and objectives

- To identify how the qualifications are devised.
- To review the content of the qualification.
- To explore how to plan the course and/or lessons.
- To understand the assessment of the qualification and how to prepare students.
- To identify the support available from Pearson.

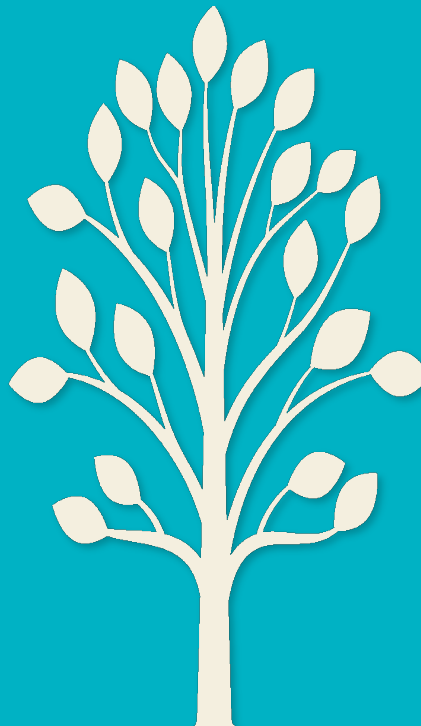


Agenda

Time (minutes)	Item
5	Welcome
5	Aims and agenda
5	Devising a qualification
5	Overview of qualification
20	Qualification content
15	Activity 1 – Identify the topic
5	Planning for teaching the course
15	Activity 2 – Order the spec points
10	Assessment
10	Activity 3 – Identify the paper
10	Preparing students for exams
10	Available support
5	Questions



Devising a qualification



Devising a qualification

UK Department for Education (DfE) publish [subject content criteria](#)

- Must be met by every GCSE qualification in Computer Science.

Panel of experts review subject content, prior learning at Key Stage 3, and next steps at Post-16.

Iterative refinement process:

- Panel of qualification design and subject experts develop assessment model.
- Panel of qualification design and subject experts develop specification content.
- Research team consult centres, teachers (secondary, FE, and HE), and industry experts.



Devising a qualification

Iterative process

- Sample assessment materials are developed by qualification design and subject experts.
- Papers are sat by subject-specific reviewers and assessed by development team.
- Research team consults centres, teachers. Students sit prototype papers.
- Ancillary documents, such as conduct of exams and submission forms, are developed.

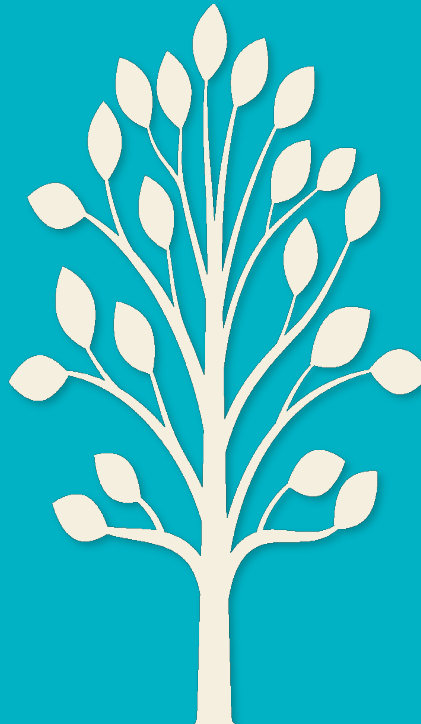
Final sign-off

- UK GCSE qualifications go to the UK Office of Qualifications and Examinations Regulation (Ofqual).
 - Specification, assessment strategies, and sample assessment materials are submitted to OfQual for accreditation.
- International GCSE qualifications go to internal sign-off.
 - Specification, assessment strategies, and sample assessment materials are submitted to an internal review for final sign-off.

Note: The exact sequence of and repetition of steps varies across qualifications. The description given here is generic.



Overview of qualification



International GCSE Computer Science

Content

1. Problem solving
2. Programming
3. Data
4. Computers
5. Communication and the internet
6. The bigger picture

Assessment Objectives (AO)

AO1 – Demonstrate knowledge and understanding of the key principles of computer science

AO2 – Apply knowledge and understanding of key concepts and principles of computer science

AO3 – Analyse problems in computational terms:

- to make reasoned judgements
- to design, program, evaluate and refine solutions

Structure of Assessment

100% external assessment

Fully linear

9 – 1 grading scale

Two papers each with a weighting of 50%

Written examination

Principles of Computer Science

Practical examination

Application of Computational Thinking

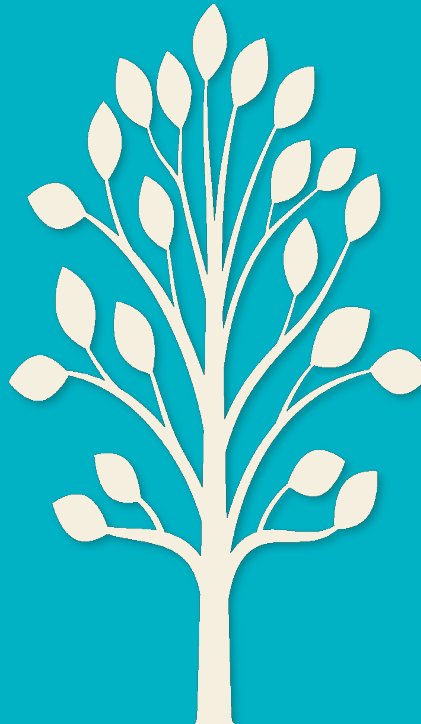


Where are each of the AOs assessed?

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%



Qualification content



Topic 1: Problem solving

Algorithms

- Create.
- Determine correct output for a given set of data.
- Standard algorithms.
- Fitness for purpose.
- Expressed in words, flowcharts, pseudocode, or program code.

Decomposition and abstraction

- Analyse problems.
- Design solutions.
- Decompose into smaller sub-problems.
- Abstraction to model real world.
- Create abstractions of real-world examples.

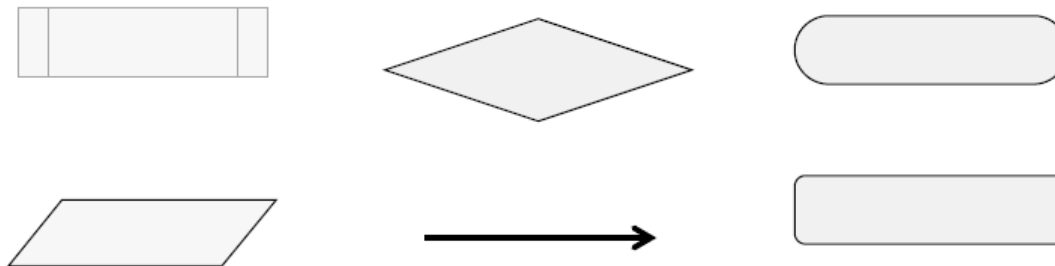


Annotation formats

Pseudocode

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

Flowcharts



Topic 2: Programming

Develop code

- Write programs, identify and fix errors, trace, evaluate.

Constructs

- Sequence, selection, repetition, iteration.

Data types and structures

- Primitive data types, data structures, records, 1-D and 2-D arrays.

Input/output

- Accept and respond to input, validation, read/write text files.

Operators

- Boolean, relational, arithmetic.

Subprograms

- Use and write, pass parameters, return results.



Topic 3: Data

Binary

- Positive, negative, conversions, addition.

Data representation

- ASCII, Unicode, image, sound.

Data storage and compression

- IEC units, Run Length Encoding, storage requirements.

Encryption

- Pigpen cipher, Caesar cipher, Vigenère cipher, Rail Fence cipher.



Topic 4: Computers

Machines and computational modelling

- Input, process, output, multi-agent, parallel.

Hardware

- Microprocessor, cache, RAM, cloud.

Logic

- AND, OR, NOT.

Software

- Operating system, utility, applications.

Programming languages

- High-level, low-level, translators.



Topic 5: Communication and the internet

Networks

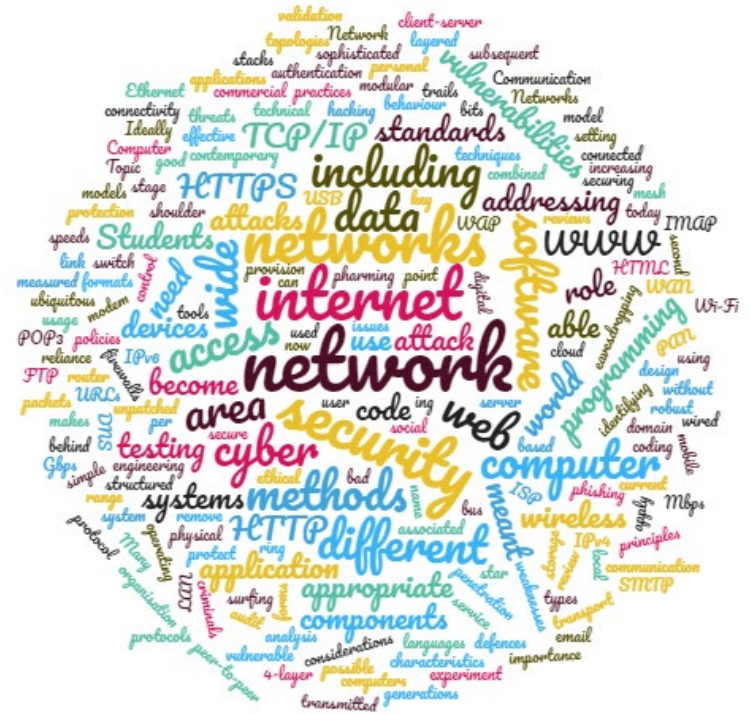
- Topologies, layered protocols, media.

Network security

- Vulnerabilities, protection, cloud.

The internet and the World Wide Web

- IP addressing, DNS, components.



Topic 6: The bigger picture

Environmental impact

- Health, energy use, resources.

Ethical impact

- Privacy, inclusion, professionalism.

Legal impact

- Intellectual property, patents, licensing, and cyber-security.

Emerging trends

- Quantum computing, DNA computing, artificial intelligence, and nanotechnology.



Activity 1

Identify the topic

Activity 1 – Identify the topic

- Here are some questions that have been created by a teacher for use in the classroom.
- For each question:
 - Identify the specification bullet point that you think it belongs to.
 - Make notes in the last column to answer the question.

Activity 1 – Identify the Topic

Here are some questions. These questions have been created by a teacher for use in the classroom.

For each question, identify the specification bullet point that you think it belongs to. Make notes in the last column to answer the question. The first one has been done for you.

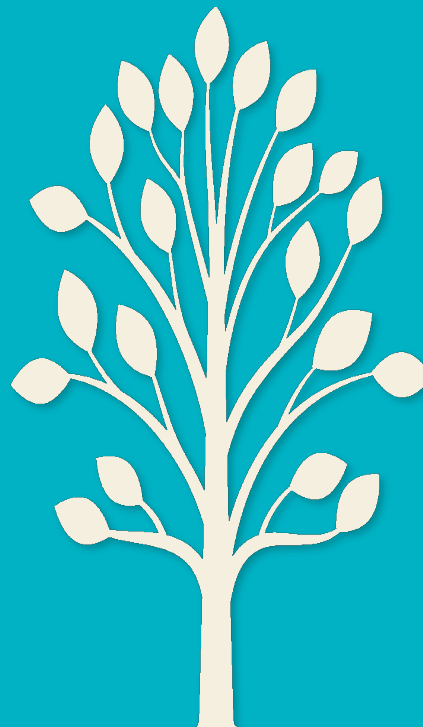
	Question	Bullet	Notes
1	Convert 4 gibibytes to kibibytes.	3.3.1	$4 \times 2^{30} / 2^{10}$
2	Discuss the statement “Artificial intelligence is more than just robots”.		
3	Give 2 advantages of IPv6 over IPv4.		
	Here is the text representation of a binary pattern stored		



	Question	Bullet	Notes
1	Convert 4 gibibytes to kibibytes.	3.3.1	$4 * 2^{30} / 2^{10}$
2	Discuss the statement 'Artificial intelligence is more than just robots'.	6.1.4	Natural language processing, machine learning, speech recognition.
3	Give 2 advantages of IPv6 over IPv4.	5.3.3	More addresses. Works better on mobile devices.
4	Here is the text representation of a binary pattern stored in memory: AABBBCDEFGGGGHIJKLMNOPQRSTUVWXYZ Explain the effect of applying a run-length encoding (RLE) algorithm to this pattern.	3.3.3	The encoded output would be longer than this input because every 'single' letter would become a 'letter, digit' pair. Only runs of 3 or more letters would compress.
5	The stored program concept assumes that the bit patterns stored in memory can represent only two items. Name these two items.	4.2.3	Data. Instructions.
6	What is the minimum number of passes that a bubble sort must make through the data, and why?	1.1.8	One. A single pass is needed, without swaps, to make sure the list is sorted. If the input is already sorted, it still takes one pass.
7	Compare the use of variables and constants in coded programs.	2.3.4	Constants are created and set once for the life of the program. Value of variables can change at any time.



Planning for teaching the course



High-level planning

- 120–140 guided learning hours.
- Lay out the teaching timeline for the whole course at the same time:
 - Ensures coverage of all topics and sub-topics.
- Weave the teaching of programming through the entire course length:
 - With two lessons per week, one could be programming related; the other could be the theoretical topics.
 - Students need repeated programming practice.
- Interleave practical computer-based activities, with traditional paper-based activities:
 - You may find it is possible to teach lessons without a computer at all.
- Leave time for revision near the exam date.

2-year course planner for Pearson International GCSE Computer Science (B - 1)																
Weeks	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	
Year 1, Term 1 (15 weeks)																
Lesson 1	Intro		Problem solving & programming				Computers		Representations of numbers			Programming languages		Hardware		OS
Lesson 2	Problem solving & programming														OS	
Weeks																
Year 1, Term 2 (12 weeks)																
Lesson 1	Networks							Boolean logic			Data rep. text					
Lesson 2	Problem solving & programming				The bigger picture			Problem solving & programming			The bigger picture		Networks			
Weeks																
Year 1, Term 3 (14 weeks)																
Lesson 1	Data rep. images		Data rep. sound		Hardware: external components				Comput. models		Network security					
Lesson 2	Problem solving & programming												Network security			
Weeks																
Year 2, Term 1 (15 weeks)																
Lesson 1	Photo shop & iStock		Data storage and compression			Hardware: secondary storage			Internet & WWW			Encryption		Embedded systems		
Lesson 2	Problem solving & programming												Encryption		Bio/Phys	
Weeks																
Year 2, Term 2 (16 weeks)																
Lesson 1	Bigger picture: Emerging trends					Topic 1: Problem solving					Topic 4: Computers					
Lesson 2	Problem solving & programming															Topic 2: Programming
Weeks																
Year 2, Term 3 (16 weeks)																
Lesson 1	Bigger picture: Emerging trends					Topic 1: Problem solving					Topic 4: Computers					
Lesson 2	Problem solving & programming															Topic 2: Programming
Weeks																
Year 2, Term 4 (16 weeks)																
Lesson 1	Bigger picture: Emerging trends					Topic 1: Problem solving					Topic 4: Computers					
Lesson 2	Problem solving & programming															Topic 2: Programming
Weeks																
Year 2, Term 5 (16 weeks)																
Lesson 1	Bigger picture: Emerging trends					Topic 1: Problem solving					Topic 4: Computers					
Lesson 2	Problem solving & programming															Topic 2: Programming
Weeks																
Year 2, Term 6 (16 weeks)																
Lesson 1	Bigger picture: Emerging trends					Topic 1: Problem solving					Topic 4: Computers					
Lesson 2	Problem solving & programming															Topic 2: Programming
Weeks																
Year 2, Term 7 (16 weeks)																
Lesson 1	Bigger picture: Emerging trends					Topic 1: Problem solving					Topic 4: Computers					
Lesson 2	Problem solving & programming															Topic 2: Programming
Weeks																
Year 2, Term 8 (16 weeks)																
Lesson 1	Bigger picture: Emerging trends					Topic 1: Problem solving					Topic 4: Computers					
Lesson 2	Problem solving & programming															Topic 2: Programming
Weeks																
Year 2, Term 9 (16 weeks)																
Lesson 1	Bigger picture: Emerging trends					Topic 1: Problem solving					Topic 4: Computers					
Lesson 2	Problem solving & programming															Topic 2: Programming
Weeks																
Year 2, Term 10 (16 weeks)																
Lesson 1	Bigger picture: Emerging trends					Topic 1: Problem solving					Topic 4: Computers					
Lesson 2	Problem solving & programming															Topic 2: Programming
Weeks																
Year 2, Term 11 (16 weeks)																
Lesson 1	Bigger picture: Emerging trends					Topic 1: Problem solving					Topic 4: Computers					
Lesson 2	Problem solving & programming															Topic 2: Programming
Weeks																
Year 2, Term 12 (16 weeks)																
Lesson 1	Bigger picture: Emerging trends					Topic 1: Problem solving					Topic 4: Computers					
Lesson 2	Problem solving & programming															Topic 2: Programming
Weeks																
Year 2, Term 13 (16 weeks)																
Lesson 1	Bigger picture: Emerging trends					Topic 1: Problem solving					Topic 4: Computers					
Lesson 2	Problem solving & programming															Topic 2: Programming
Weeks																
Year 2, Term 14 (16 weeks)																
Lesson 1	Bigger picture: Emerging trends					Topic 1: Problem solving					Topic 4: Computers					
Lesson 2	Problem solving & programming															Topic 2: Programming
Weeks																
Year 2, Term 15 (16 weeks)																
Lesson 1	Bigger picture: Emerging trends					Topic 1: Problem solving					Topic 4: Computers					
Lesson 2	Problem solving & programming															Topic 2: Programming
Weeks																
Year 2, Term 16 (16 weeks)																
Lesson 1	Bigger picture: Emerging trends					Topic 1: Problem solving					Topic 4: Computers					
Lesson 2	Problem solving & programming															Topic 2: Programming
Weeks																
Year 2, Term 17 (16 weeks)																
Lesson 1	Bigger picture: Emerging trends					Topic 1: Problem solving					Topic 4: Computers					
Lesson 2	Problem solving & programming															Topic 2: Programming
Weeks																
Year 2, Term 18 (16 weeks)																
Lesson 1	Bigger picture: Emerging trends					Topic 1: Problem solving					Topic 4: Computers					
Lesson 2	Problem solving & programming															Topic 2: Programming
Weeks																



This planner is in your delegate pack: S08 iGCSE CompSci Year planner v1.pdf

Teaching support materials

- Schemes of work, outlined by term:
 - Scheme of work document.
 - Lesson slides.
 - Student activities.
 - Solutions.
- Sample lesson plans.



Scheme of work - 2017
Year 1, Term 1

Scheme of work - 2017
Year 1, Term 2

Scheme of work - 2017
Year 1, Term 3



Int.GCSE_COMPSCI_Y1_T1_Act_Solutions.docx



InternationalGCSE_COMPSCI_Y1_T1_Activities.doc



InternationalGCSE_COMPSCI_Y1_T1_SOW.docx



Year 1, Term 1 Slides.pptx



Computer Science Lesson Plans
| DOCX 189.9 KB | 31 Aug 2017

Specification and sample
assessments (3)

Exam materials (8)

Forms and administration
(3)

Teaching and learning
materials (16)



Activity 2

**Order the spec
points**



Pearson
Edexcel

Activity 2 – Order the specification points

You are going to:

- Analyse the contents from topics across the specification.
 - Evaluate each statement.
 - Identify the specification point.
 - Put items into **a teaching order**.
-
- Keep in mind that the numbering of the bullets may not represent the most suitable teaching order.

Activity 2 – Put into teaching order

In this activity, you are going to analyse the contents from topics across the specification. Evaluate each statement, identify the specification point, and put them in a teaching order. Keep in mind that the numbering of the bullets presented in the specification may not represent the most suitable teaching order.

These points have been chosen because they represent a basic understanding of how the machine works. This will create a good foundation for making the connection between the abstract world of programming and the world of hardware.

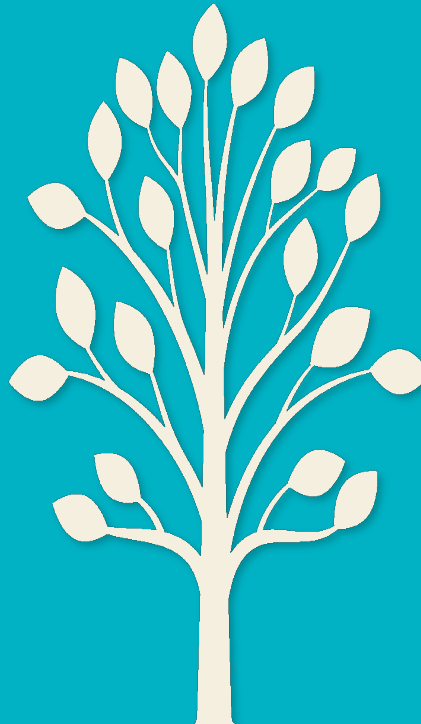
Order	Spec	Description
		Understand the concept of a stored program and the role of components of the CPU (control unit (CU), arithmetic/logic unit (ALU), registers.



Order	Spec	Description
4	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).
9	4.3.2	Be able to produce logic statements for a given problem.
2	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.
10	3.2.1	Understand how computers encode characters using ASCII and Unicode.
5	3.1.1	Understand that computers use binary to represent data (numbers, text, sound, graphics) and program instructions.
7	3.1.3	Be able to convert between binary and denary whole numbers (0–255).
8	4.3.1	Be able to construct and interpret truth tables for a given logic statement (AND, OR, NOT).
3	4.2.2	Understand the function of different types of memory (random-access memory (RAM), read-only memory (ROM), cache, virtual memory).
6	3.1.2	Understand how computers represent and manipulate numbers (unsigned integers, signed integers (sign and magnitude, two's complement)).
1	4.1.1	Understand the input-process-output model.



Assessment



Paper 1: Principles of Computer Science

Write your name here							
Surname	Other names						
Pearson Edexcel Level 1/Level 2 International GCSE (9–1)	Centre Number <table border="1"><tr><td></td><td></td><td></td><td></td><td></td><td></td></tr></table>						
	Candidate Number <table border="1"><tr><td></td><td></td><td></td><td></td><td></td><td></td></tr></table>						
Computer Science Paper 1: Principles of Computer Science							
Sample assessment material for first teaching September 2017 Time: 2 hours	Paper Reference 4CP0/01						
You must have: A pseudocode reference	Total Marks <table border="1"><tr><td></td></tr></table>						

- 2-hour, non-tiered practical examination.
- All candidates sit examination at the same time.
- 80 marks in total.
- Traditional written examination paper.
- All questions require a written response on the paper.
- All questions are mandatory.
- Question types include multiple-choice, short open-response, open-response and extended open-response.



Paper 2: Application of Computational Thinking

Write your name here	
Surname	Other names
Pearson Edexcel	Centre Number
Level 1/Level 2	Candidate Number
International GCSE (9–1)	
Computer Science	
Paper 2: Application of Computational Thinking	
Sample assessment material for first teaching September 2017 Time: 3 hours	Paper Reference 4CP0/02
You must have: A computer workstation with appropriate programming language code editing software and tools, including a code interpreter/compiler, CODES folder containing code and data files, pseudocode reference	Total Marks

- 3-hour, nontiered practical examination.
- 3-day window.
- 80 marks total.
- All questions are mandatory.
- The paper consists of multiple-choice, short open-response, open-response, extended open response answer and task-based questions.



Paper 2: Application of Computational Thinking

Write your name here			
Surname		Other names	
Pearson Edexcel Level 1/Level 2 International GCSE (9–1)		Centre Number	Candidate Number
Computer Science Paper 2: Application of Computational Thinking			
Sample assessment material for first teaching September 2017 Time: 3 hours		Paper Reference 4CP0/02	
You must have: A computer workstation with appropriate programming language code editing software and tools, including a code interpreter/compiler, CODES folder containing code and data files, pseudocode reference			Total Marks

- Three programming languages (Python, C# or Java).
- Candidates must have:
 - A computer workstation with one of the approved programming languages, editing software and tools, including a code translator.
 - Access to a CODES folder containing the supplied code and data files.
 - A pseudocode reference sheet.
- No internet access.
- A pseudocode reference document will be available for learners to reference during the assessment.



Activity 3

**Identify the
Paper**



Pearson
Edexcel

Activity 3 – Identify the Paper

- Here are some questions that have been created by a teacher for use in the classroom.
- You are going to:
 - Analyse the given question.
 - Identify the specification point.
 - Identify if the question is likely to be in Paper 1 or Paper 2.
 - Make notes in the last column about how you would answer the question.

Activity 3 – Identify the Paper

In this activity, you are going to analyse some questions to decide if they are for Paper 1: Fundamentals of Computational Thinking or Paper 2: Application of Computational Thinking.

Evaluate each question, identify the specification point, and identify the paper it is likely to be in.

The questions in this activity have not been developed by examiners, so they are not guaranteed to appear in either paper, depending on the content of the examinations. The questions may also appear in either paper, depending on the content of the examinations.

Paper	Spec	Question
		Here is an array of 52 strings: [A, AA, B, BB, C, CC, ..., Y, YY, Z, ZZ]. You want to find the string MM. State an appropriate search algorithm to use.
		Give two advantages of using a compiler over an interpreter to translate a programming language.



Note: This activity extends over two pages in the delegate booklet.

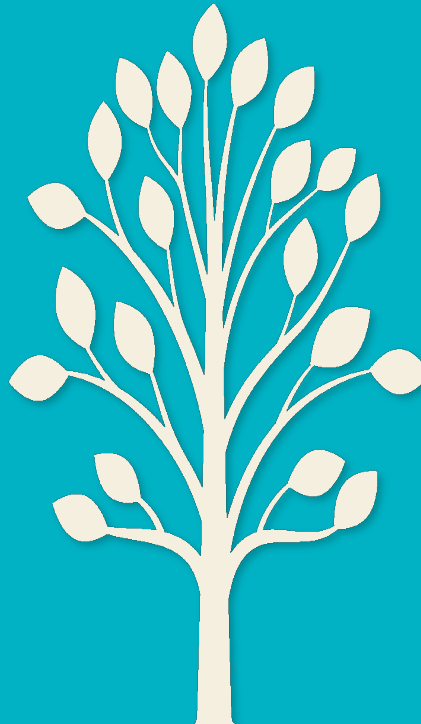
Paper	Spec	Question
1	1.1.7	Here is an array of 52 strings: [A, AA, B, BB, C, CC, ... X, XX, Y, YY, Z, ZZ]. You want to find the string MM. State and justify an appropriate search algorithm to use.
1	4.5.2	Give two advantages of using a compiler over an interpreter to translate a programming language.
2	2.3.4	Explain why programmers use constants in their code.
1	5.1.6	State the names of the four-layer TCP/IP model in order from lowest to highest.
2	1.1.2	Here is an example of a 'for' loop: FOR index FROM 1 TO 10 DO SEND <u>testScores[index]</u> TO DISPLAY
		END FOR Rewrite this logic to use a 'while' loop for any number of scores in the data structure <u>testScores</u> . Use your chosen programming language.



1	3.2.1	In the ASCII table, the hexadecimal code 4B represents the letter K. What is the hexadecimal code for the letter S?
1	6.1.4	Discuss this statement: DNA computing will allow us to store much more information in the future than we can store now.
2	2.6.2	Write a subprogram that takes three integers as parameters and returns their average. Use your chosen programming language.



Preparing students for exams



Preparation

All topics

- Flashcards.
- Fill in the blanks.
- Multiple-choice questions.
- Classification into categories.
- Putting in order.
- Creating or completing drawings.
- Odd one out.
- Correct the incorrect response.

Programming-related topics

- Reading, creating, and amending algorithms written in pseudocode or flowcharts.
- Comparison and contrasting characteristics.
- Finding and correcting errors in pseudocode and flowcharts.
- Peer instruction to encourage students to learn from each other.



Coding preparation

- Finding and correcting errors in code, using an IDE.
- PRIMM (predict, run, investigate, modify, make).
- Paired programming to write and debug code.
- Faded worked examples (scaffolding is removed over time).
- Peer instruction to encourage students to learn from each other.
- Parsons problems, where code must be arranged in order.
- Understanding requirements and writing code to meet them.



Exam tips

Both papers:

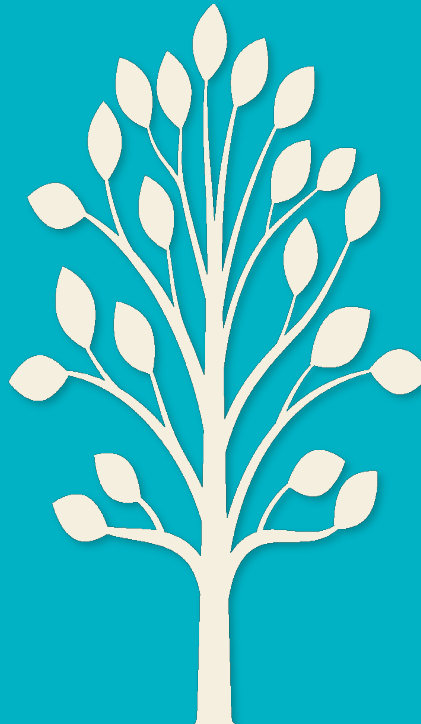
- Students should practise with full papers:
 - Be familiar with layout.
 - Be able to judge timings.
- Attempt every question, partial marks are awarded.
- Watch the time to make sure there is the opportunity to respond to every question.
- Write legibly; printing is acceptable.
- Use subject-specific terminology accurately.
- Read the question carefully, highlight important requirements.
- Sufficient space is provided on the papers, students don't need to fill the whole booklet.

Practical paper:

- Be confident in the use of your programming language development environment.
- Make a backup copy of the original files, to ensure starting from a clean state, if required.
- Save your working files often, in case of a technical fault.



Available support



Support



Getting Started Guide

Includes mapping of changes, content and assessment guidance, course planer and resource list.

Editable Scheme of Work

Includes activities to support transferable skills development.

Exemplar marked responses

Marked student responses to questions on SAMs or past papers.

Subject transferable skills

Identifies skills that will help students cope with the demands of further study and ultimately into employment.



Support

Getting Ready to Teach Events

May be face to face, online, or recorded events where trainers will guide you to understand the qualification and how to prepare students.

ResultsPlus

Free online service giving instant and detailed analysis of your students' exam and mock performances.

Regional Support Manager

Access to a regionally based support manager for any query.

Subject Advisor

For any subject-related query you have. You can also sign up to the mailing list.



Subject advisor



Tim Brady

Computer Science and ICT Subject Advisor



UK: 0333 016 4160

Intl: +44 (0)333 016 4160

[➤ Contact us](#)

Please have a look at your new community



@pearsonICT

@Pearson_CS



BTEC Tech Award in DIT Facebook group

BTEC Firsts in I&CT Facebook group

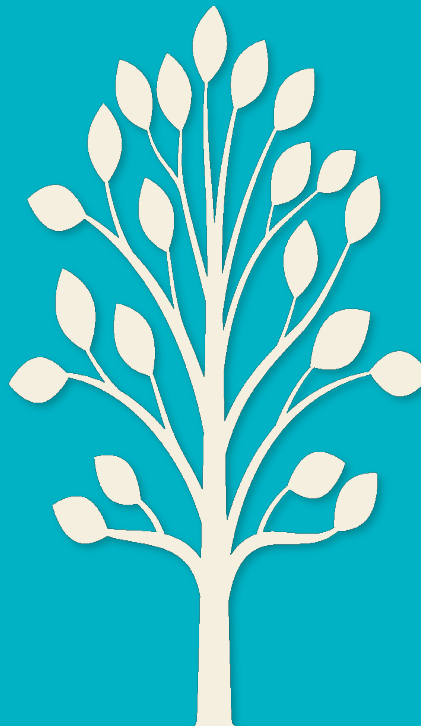
CiDA/DiDA Facebook group

GCSE Computer Science Facebook group



Pearson
Edexcel⁴⁴

Any questions?



Thank you for attending.

ALWAYS LEARNING