

EDEXCEL INTERNATIONAL GCSE (9-1)

Computer Science

GETTING READY TO TEACH

Event code: 17IOAN01

First teaching in 2017, first assessment in 2019.



Your Online Environment

XX Technical Difficulties & Support

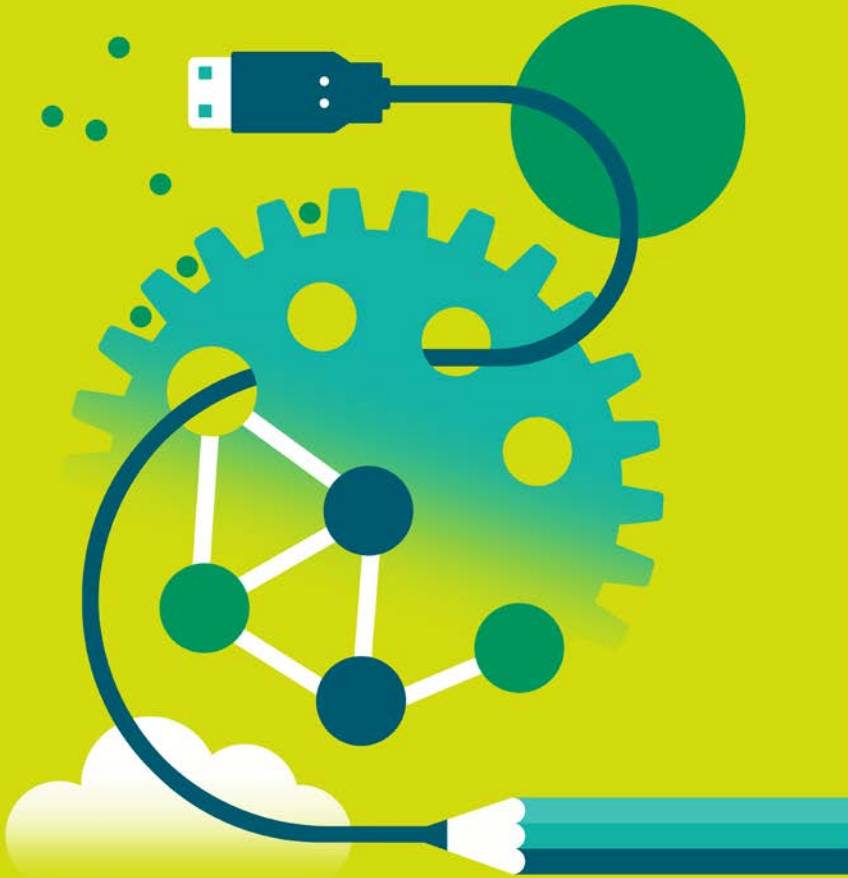
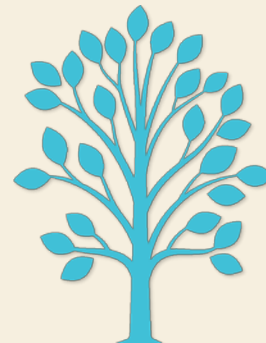
XX Recording

XX Communication in and online environment

XX Asking questions

XX Using polls

XX Downloading Documents



Aims and Objectives

To equip you with the information you need to successfully plan and deliver international GCSE Computer Science, including:

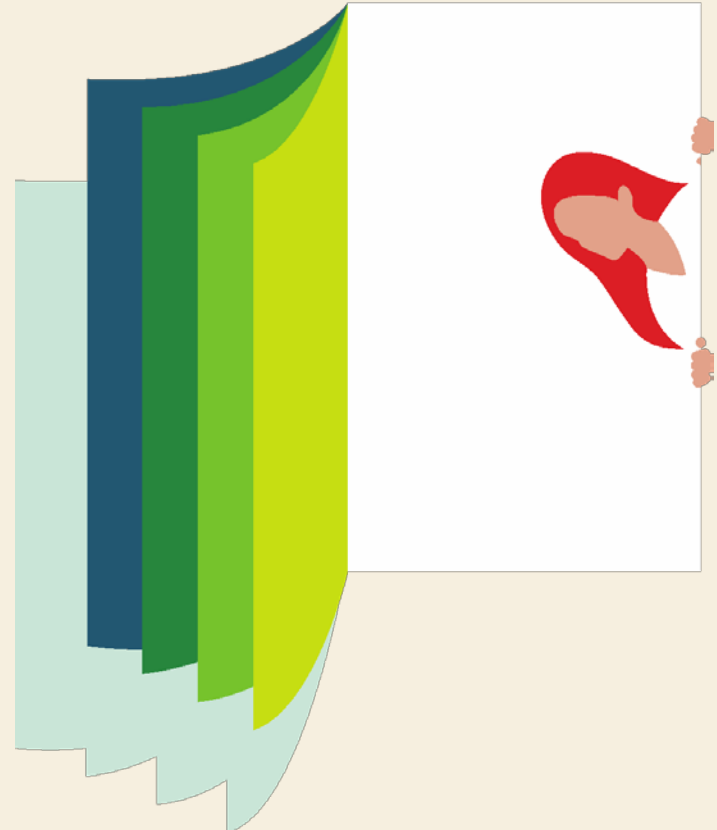
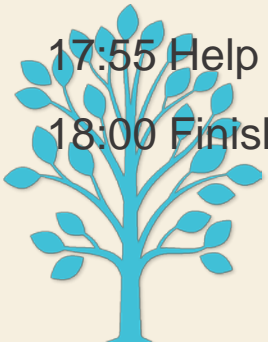
- XX** the structure and content of the qualification
- XX** how it is assessed, including requirements for the practical examination
- XX** support available from Pearson and others
- XX** teaching and delivery strategies

And to give you an opportunity to share ideas and pool your expertise



Agenda

- 16:00 Welcome and introductions
- 16:05 Overview of the qualification
- 16:15 Problem solving and programming
- 16:40 Paper 02 – Application of Computational Thinking
- Discussion and short break
- 17:05 Topics 3 – 6
- 17:30 Paper 01 – Principles of Computer Science
- 17:40 Planning and Delivery
- 17:55 Help and Support from Pearson
- 18:00 Finish

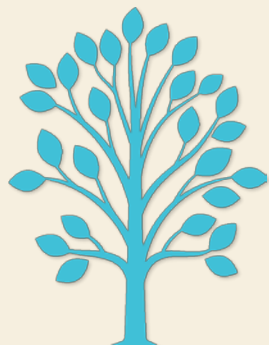


What is Computer Science?



Key Features

- Engaging, contemporary content
- Focus on computational thinking
- Examination-only assessment
- Clear and straight-forward question papers
- Choice of three programming languages
- Fosters progression



Overview of the qualification

Content

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

Assessment Objectives

- AO1.** Demonstrate knowledge and understanding of 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

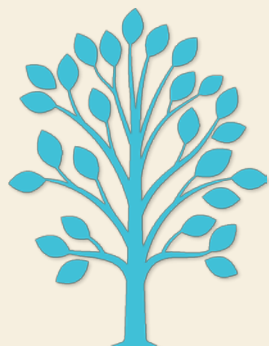
Two papers each with a weighting of 50%

Written examination

Principles of Computer Science

Practical examination

Application of Computational Thinking



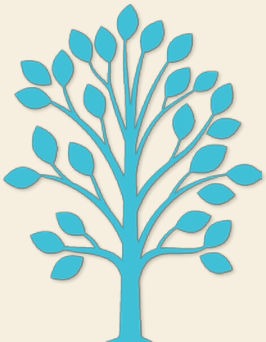
9-1 grading scale

Awarding

- The grading system is changing but our commitment to awarding grades that accurately reflect learner exam performance remains the same.
- We set new grade boundaries (minimum number of marks needed to achieve each grade) for each assessment of each qualification.

Benefits

- Greater differentiation across levels of attainment e.g. 2 grades where the current C grade is
- Rewards truly outstanding achievement with the grade 9
- Provides more information about student attainment to help progression to A Level
- Same scale for Pearson Edexcel GCSE and International GCSE allows clear comparison with English standards, unlike old A* to G grading



Problem solving and programming:

**What students must understand and
be able to do**



Problem solving and programming

```

FUNCTION calc_averages(data)
# calculates average temp and wind speed for selected data
BEGIN FUNCTION
SET totalTemp TO 0
SET totalWindSpeed TO 0
SET numReadings TO LENGTH(data)
FOR EACH item FROM data
SET totalTemp TO totalTemp + data[3]
SET totalWindSpeed TO totalWindSpeed + data[4]
ENDFOR
SET averageTemp TO totalTemp/numReadings
SET averageWindSpeed TO totalWindSpeed/numReadings
RETURN averageTemp, averageWindSpeed
END FUNCTION

```

```

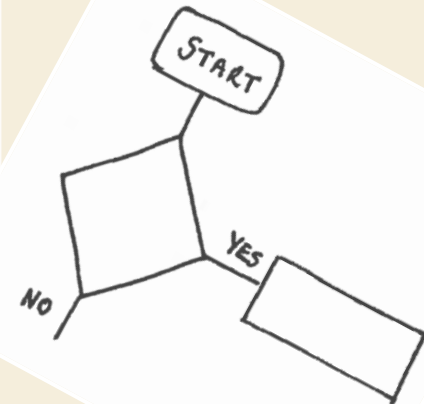
theArtists = [
    ["Andy", "Warhol", 1928],
    ["Pablo", "Picasso", 1881],
    ["Salvador", "Dali", 1904],
    ["Lavinia", "Fontana", 1552],
    ["Jackson", "Pollock", 1912],
    ["Henri", "Matisse", 1869],
    ["Frida", "Kahlo", 1907],
    ["Georgia", "O'Keeffe", 1887],
    ["Kara", "Walker", 1969],
    ["Yayoi", "Kusama", 1929]
]

theLabels = [] # Put the new user labels into this structure

# Make the artist labels
for person in theArtists:
    newRecord = person[1][0] + person[0][0] + str(person[2])
    theLabels.append(newRecord)
print ("The new userIDs are: ", theLabels)

# Find and print the youngest person and their birthdate
maxDate = 0
for person in theArtists:
    if person[2] > maxDate:
        maxDate = person[2]
        maxPerson = person
print (maxPerson[0], maxPerson[1], "is youngest", str(maxPerson[2]))

```



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

Instructions

- Use **black** ink or ball-point pen.
- **Fill in the boxes** at the top of this page with your name, centre number and candidate number.
- Answer **all** questions.
- Answer the questions **requiring a written answer** in the spaces provided – *there may be more space than you need.*
- Only **one** programming language must be used throughout the test.
- Carry out practical tasks on the computer system and save new or amended code using the name given with the appropriate file extension.
- Do **not** overwrite the original code and data files provided to you.
- You must **not** use the Internet during the test.

Information

- The total mark for this paper is 80.
- The marks for **each** question are shown in brackets
- This paper covers Python, C# and Java.
- The CODES folder in your user area includes all the code and data files you need.
- The invigilator will tell you where to store your work.

Advice

- Read each question carefully before you start to answer it.
- Save your work regularly.
- Check your answers and work if you have time at the end.

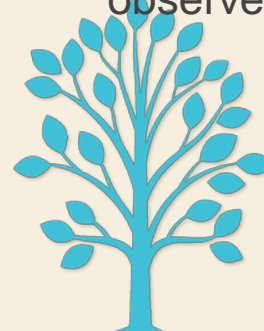
Turn over ►

S54964A



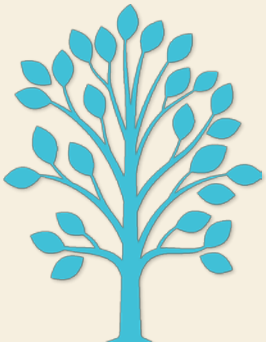
©2017 Pearson Education Ltd.
1/1/1

- 3 hour, untiered practical exam
- 5-day window
- Candidates must have:
 - a computer workstation with one of the approved programming language (Python, C# or Java),
 - editing software and tools, including a code translator
 - access to a CODES folder containing the supplied code and data files
 - a pseudocode reference sheet
- Some questions require a written response
- No Internet access
- No choice of questions.
- File naming conventions must be observed



Paper 02 question-types

- Answer questions about a piece of code, e.g. Q1(a), Q1(b)i), Q3(b)
- Answer a 'theory' question not directly related to any code, e.g. Q1(b)(ii), Q1(c)(i), Q2(b), Q3(a), Q4(a), Q4(b), Q4(c)
- Amend a piece of code to correct errors, e.g. Q1(c)(ii)
- Fill in the gaps in a piece of code, e.g. Q2(c)
- Generalise a piece of code
- Determine what value a variable will hold at a given point in a program, e.g. Q5(a)
- Convert an algorithm into executable code, e.g. Q2(a)
- Write a program from scratch, e.g. Q3(c), Q5(b), Q6



Conducting the practical exam

- Set up a separate secure user area with sufficient storage for each candidate
- Download the data files, check for compatibility and copy into each candidates' user area
- Check that the computer equipment and software to be used is suitable
- Ensure that at least one invigilator is able to deal with any technical issues that may arise
- Ensure that during the examination candidates :
 - ✓ can only access the files required for the examination
 - ✓ cannot access the internet or refer to textbooks
 - ✓ cannot save the files they produce in a central, unsecure location or on a portable storage device
 - ✓ cannot view each other's screens
 - ✓ can make use of offline help facilities /software-specific manuals
- Return paper scripts to Pearson and upload digital responses
- Refer to the ICE document for further, detailed instructions



Discussion point

**What's the best way to
teach problem-solving
and programming?**



Free learning to program resources

Codecademy: <https://www.codecademy.com>

Grok Learning: <https://groklearning.com>

Sentdex Youtube channel: <https://www.youtube.com/user/sentdex/videos>

Khan Academy: <https://www.khanacademy.org>

Interactive tutorials: <https://learnpython.org>, <https://learnjavaonline.org>,
<https://www.learncs.org>

CodingBat: <http://codingbat.com>

Code.org hour of code: <https://code.org>

Invent your own computer games with Python:
<http://inventwithpython.com/chapters/>

Hacking secret ciphers with Python: <http://inventwithpython.com/hacking/>



Concepts and principles of Computer Science

3. Data

- Binary
- Data representation
- Data storage and compression
- Encryption

4. Computers

- Machines and computational models
- Hardware
- Logic
- Software
- Programming languages

5. Communication and the internet

- Networks
- Network security
- The internet and the world wide web

6. The bigger picture

- Emerging trends, issues and impact



Computer-related mathematics

Students must be able to:

- ✓ Convert between number bases
- ✓ Perform binary addition, division and multiplication (logical and arithmetic shifts)
- ✓ Construct and interpret expressions and logic statements
- ✓ Convert between units of measurement
- ✓ Calculate file sizes



Write your name here

Surname	Other names
---------	-------------

Pearson Edexcel
Level 1/Level 2
International GCSE (9–1)

Centre Number	Candidate Number
<input type="text"/>	<input type="text"/>

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
---	-------------

Instructions

- Use **black** ink or ball-point pen.
- **Fill in the boxes** at the top of this page with your name, centre number and candidate number.
- Answer **all** questions.
- Answer the questions in the spaces provided
– *there may be more space than you need.*

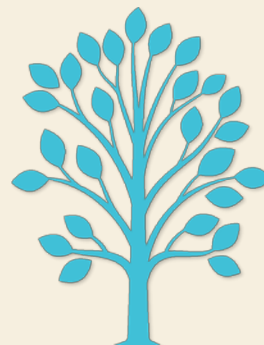
Information

- The total mark for this paper is 80.
- The marks for **each** question are shown in brackets
– *use this as a guide as to how much time to spend on each question.*
- You are not allowed to use a calculator.

Advice

- Read each question carefully before you start to answer it.
- Try to answer every question.
- Check your answers if you have time at the end.
- Marks will not be awarded for using product or trade names in answers without giving further explanation.

- 2 hour, untiered written exam
- No choice of questions
- Variety of question types, including some multiple choice
- Consistent use of command words
- At least one extended-writing question worth 6 marks
- Some questions involve working with algorithms
- Use of a calculator is not allowed
- Space for drafting where appropriate



Turn over ►



Planning the course



Assessment structure

Paper 01: Principles of Computer Science

2 hours

50%

Paper 02: Application of Computational Thinking

3 hours

50%



Subject topics by paper

Topics	Paper 01	Paper 02
1. Problem Solving	✓	✓
2. Programming	✓	✓
3. Data	✓	✓
4. Computers	✓	✓
5. Communication and the internet	✓	✓
6. The Bigger Picture	✓	✓



Considerations

- Designed to be delivered in 120 – 140 guided learning hours (approximately two 1 hour lessons a week over two years)
- Both the exams must be taken at the end of the course
- Problem solving and programming must be developed and practised throughout the course
- How much experience of programming students already have
- Students won't be using computers in every lesson
- There are opportunities to link theory and practical work



2-year course planner for Pearson International GCSE Computer Science (9 - 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	Representation of numbers	Programming languages	Hardware	OS
Lesson 2	Problem solving & programming						OS

Weeks 1 2 3 4 5 6 7 8 9 10 11 12

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 1 2 3 4 5 6 7 8 9 10 11 12 13 14

Year 1, Term 3 (14 weeks)

Lesson 1	Data rep: images	Data rep: sound	Hardware: internal components	Comput. models	Network security
Lesson 2	Problem solving & programming				Network security

Weeks 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15

Year 2, Term 1 (15 weeks)

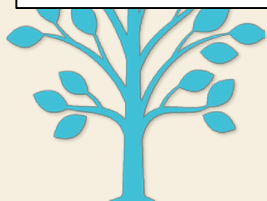
Lesson 1	Prob solv & prog	Data storage and compression	Hardware: secondary storage	Internet & WWW	Encryption	Embedded systems
Lesson 2	Problem solving & programming				Encryption	Big.Pict

Weeks 1 2 3 4

Year 2, Term 2 (4 weeks)

Lesson 1	Bigger picture: Emerging trends
Lesson 2	Problem solving & programming

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



Scheme of work materials

Week 4 Lesson	Spec ref	Lesson summary	Lesson content	Lesson resources	Transferable skills
1	3.3.2 3.3.3 3.3.4	Data storage and compression: RLE	Show the YouTube video 'Run Length Encoding Visualisation' to introduce the concept of RLE. http://www.youtube.com/watch?v=yPdNsoyym_E Give students the opportunity to experience RLE compression in action by decompressing a 4-bit colour image encoded using RLE. (Activity 4.1.1) Explain how to calculate the size (in bytes) of the uncompressed and the compressed files, pointing out the small size reduction. (Activity 4.1.2) Point out that the advantage of a smaller file is gained at the expense of the computer having to carry out more processing – the file has first to be compressed and then it has to be decompressed before it can be used. This is especially true in this case, where compression only saves one byte. Ask students to consider whether the file would compress more or less if colours were represented by 4-bit codes. Introduce students to the RLE calculator: http://mathcelebrity.com/runlengthcode.php . Give them an opportunity to try it out and get to grips with how it works. (Activity 4.1.3) Homework: Ask students to complete Activity 4.1.3	Week 4, Lesson 1 activities YouTube video: 'Run Length Encoding Visualisation' RLE calculator	Critical thinking Self-direction
2	1.1.6 3.3.3	Data storage and compression: RLE	Ask students to work in pairs to write and test an RLE algorithm they completed for the last homework. Homework: Ask students to complete the summary sheet. (Activity 4.2.2).		

Activity 4.2.1

Program solution provided in [ProgCode](#) folder.

```
#Activity 4.2.1
def run_length_encoding(data):
    rleCode = ""
    length = len(data)
    if length == 0:
        rleCode += ""
    elif length == 1:
        rleCode += str(length) + data
    else:
        count = 1
        index = 1
        while index < length:
            # Checks if it is the same letter.
            if data[index] == data[index - 1]:
                count += 1
            else:
                rleCode += str(count) + data[index - 1]
                count = 1
                index += 1
            rleCode += data[index - 1] + str(count)
        rleCode += str(count) + data[index - 1]
    return rleCode

#Main program
yourData = input("Enter the data you want to compress:")
compressed = run_length_encoding(yourData)
print("\n", compressed)
```

Activity 4.2.2 (homework)

Explain what is meant by lossless compression.	Lossless data compression reduces the size of files in such a way that the original data can be perfectly reconstructed from the compressed data – nothing is lost.
--	---

Activity 4.1.4 (homework)

- Explain how the RLE compression algorithm works.
- Here is some data used to represent an image. Each pixel is encoded as a character.
AADACCEFBAAECFGBDGE
Explain why a RLE algorithm may not be appropriate for encoding this image.
- Here is a partly completed RLE algorithm expressed as a written description. Fill in the gaps to complete it.
 - Start with the first character in the string.
 - Write down the number 1.
 - Compare the first character with the next character on the right.
 - If they are the same, _____.
 - If they are not the same, _____.
 - Move on to the next character on the right.
 - Go back to step 2 and repeat until you reach the end of the string.
 - _____.

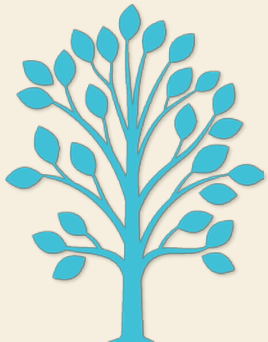
Week 4, Lesson 2 activities

Activity 4.2.1

Implement the RLE compression algorithm in Python.

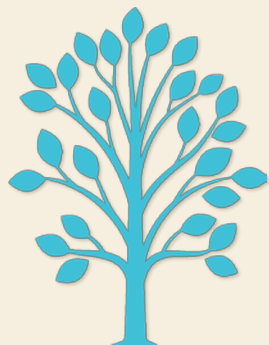
Test out your program using these data strings:

- AAAABBBBBBBBBBCADDDEEFFFFFFFF
- ABCABCABCABCABCABCABCABC



Other resources

- Computing as School (CAS)
- CS4FN (Computer Science for Fun)
- Computer Science Unplugged
- Text books for the regulated GCSE in Computer Science
- The Cybersecurity Challenge Schools Programme



INTERNATIONAL GCSE

Computer Science (9-1)

GETTING STARTED GUIDE

Pearson Edexcel International GCSE in Computer Science (4CP0)

For first teaching in September 2017

First examination June 2019



Transferable Skills

- Skills frameworks adapted to support design of new Edexcel International GCSEs
- Ensure learners acquire skills needed to access Higher Education and fulfilling careers



Cognitive skills

Core skills brain uses to think, learn and reason – used to carry out any task.



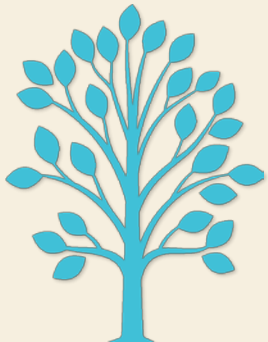
Intrapersonal Skills

Emotional intelligence, ability to know, understand and manage own emotions and learning.



Interpersonal Skills

Life skills used every day to communicate and interact with others, individually and in groups.



Subject Advisor

Tim Brady

Computer Science and
ICT Subject Advisor



Phone:

UK: 020 7010 2161

Intl: +44 (0) 20 7010 2161

Email:

TeachingICT@pearson.com

TeachingComputerScience@pearson.com

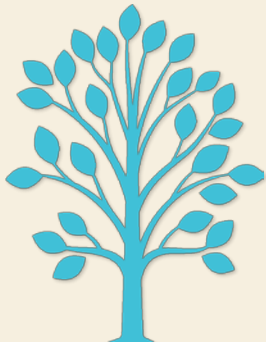
Twitter:

[@pearsonICT](https://twitter.com/pearsonICT)

[@Pearson_CS](https://twitter.com/Pearson_CS)

Facebook:

<https://www.facebook.com/groups/421769201257111/>



Useful Links

1. [Grade Boundaries](#)

This page shows the minimum marks needed to achieve a certain grade for all UK and international examinations. Also refer to the examiners report which is available for download with other documents.

2. [Examination Results Statistics](#)

Results statistics summarise the overall grade outcomes of candidates sitting Edexcel examinations.

3. [Results Plus](#)

Edexcel's free online service giving instant and detailed analysis of your students' exam and mock performance.

See your students' scores for every exam question.

Understand how your students' performance compares with Edexcel national averages.

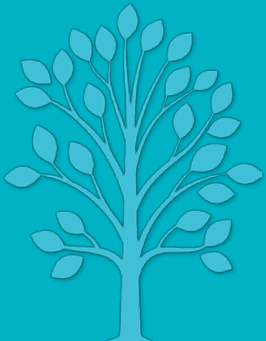
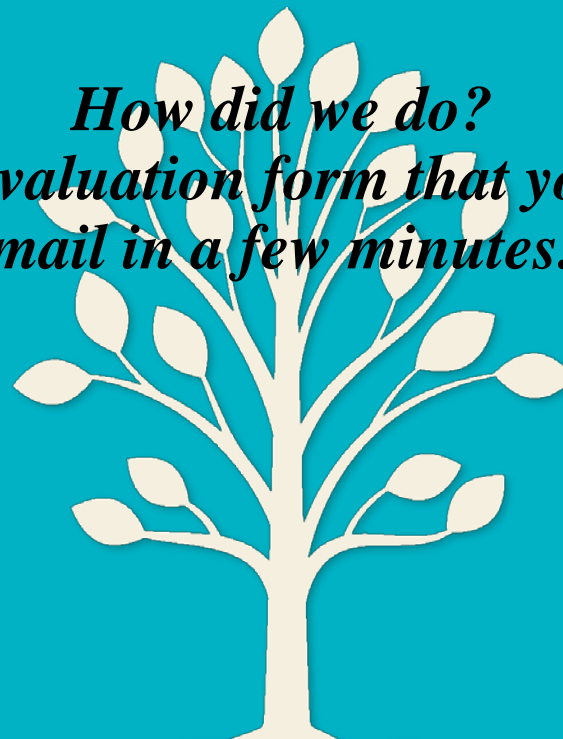


Any questions?

Thank you for attending this event.

How did we do?

Please fill in the evaluation form that you'll receive via e-mail in a few minutes.



ALWAYS LEARNING