Thank you for joining today's event

Teacher Network:
GCSE Computer Science J277

- This online session will begin at the scheduled start time
- Please ensure that you are using Chrome or Firefox as your web browser
- If you have any questions for your presenter or host, please type them into the chat window

OCR
Oxford Cambridge and RSA

Troubleshooting

Can you hear the presenter clearly?

When the poll appears on your screen, please select Yes or No using the radio buttons

Are you having issues with sound?

I can't hear anything. What should I do?

The main reason for sound issues is use of an incompatible web browser. To make sure the session runs smoothly, and for the best experience, please use **Chrome** or **Firefox**

I can't hear anything. Can I change the volume?

You can adjust your speaker volume in 'My Settings'. Select 'My Settings' from the Collaborate panel (to open, click on the purple arrow icon in the bottom right of the screen). Click 'Audio and Video Settings'

I am still having sound issues – what next?

If you continue to have issues with sound, please log out using the X button located at the top right of your screen and then re-join Blackboard using the guest link provided in your joining instructions

Contact your host

If you have tried all of the above and are still experiencing difficulty using Blackboard, please send a message to your host via the Blackboard chat window or e-mail networks@ocr.org.uk

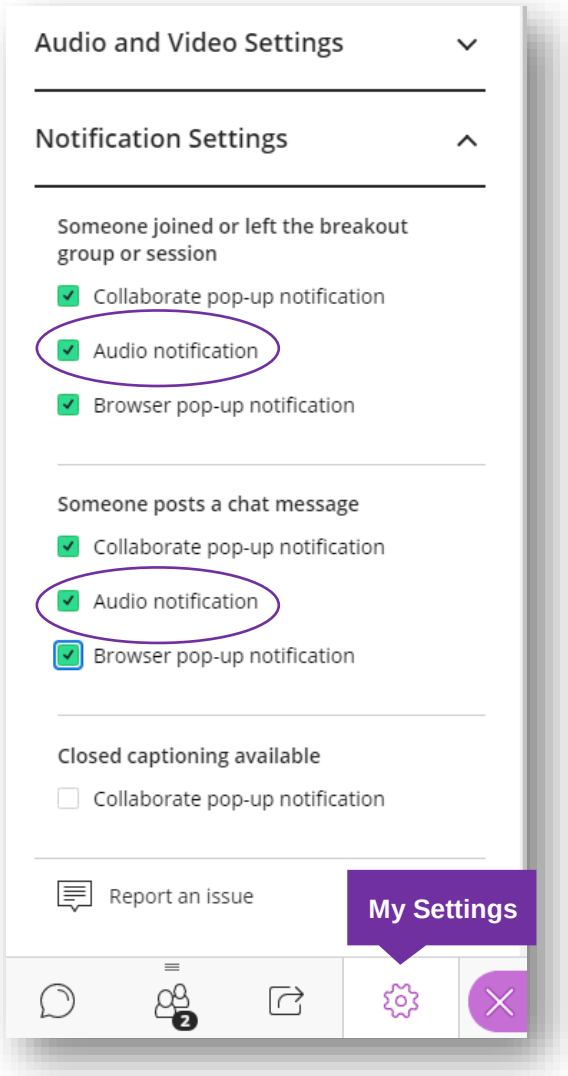
Notification settings

To avoid any distractions during the session, you may find it useful to mute notifications

Select ‘My Settings’ from the Collaborate panel

Click on ‘Notification Settings’

Untick the ‘Audio notification’ radio buttons



Teacher Networks: GCSE Computer Science J277

Summer 2023 exam

OCR
Oxford Cambridge and RSA

Exam survey

- Timing
- Support material
- Paper expectations
- Question layout
- Coding

- https://teach.ocr.org.uk/feedback-and-surveys

What was Summer 2023 like?

Discussion forum

- Please put your comments in chat

- We would like to hear about:
  - Component 2 format (A/B)
  - How the "formal response" questions went
  - Overall accessibility
  - Did you like more MCQs, Tick boxes and Fill in the blanks?
  - Did Section B replicate what you do in the classroom?
  - Any other comments/feedback to look at for the future?

Teacher Networks: GCSE Computer Science J277

Approaches for programming

OCR
Oxford Cambridge and RSA

Introduction

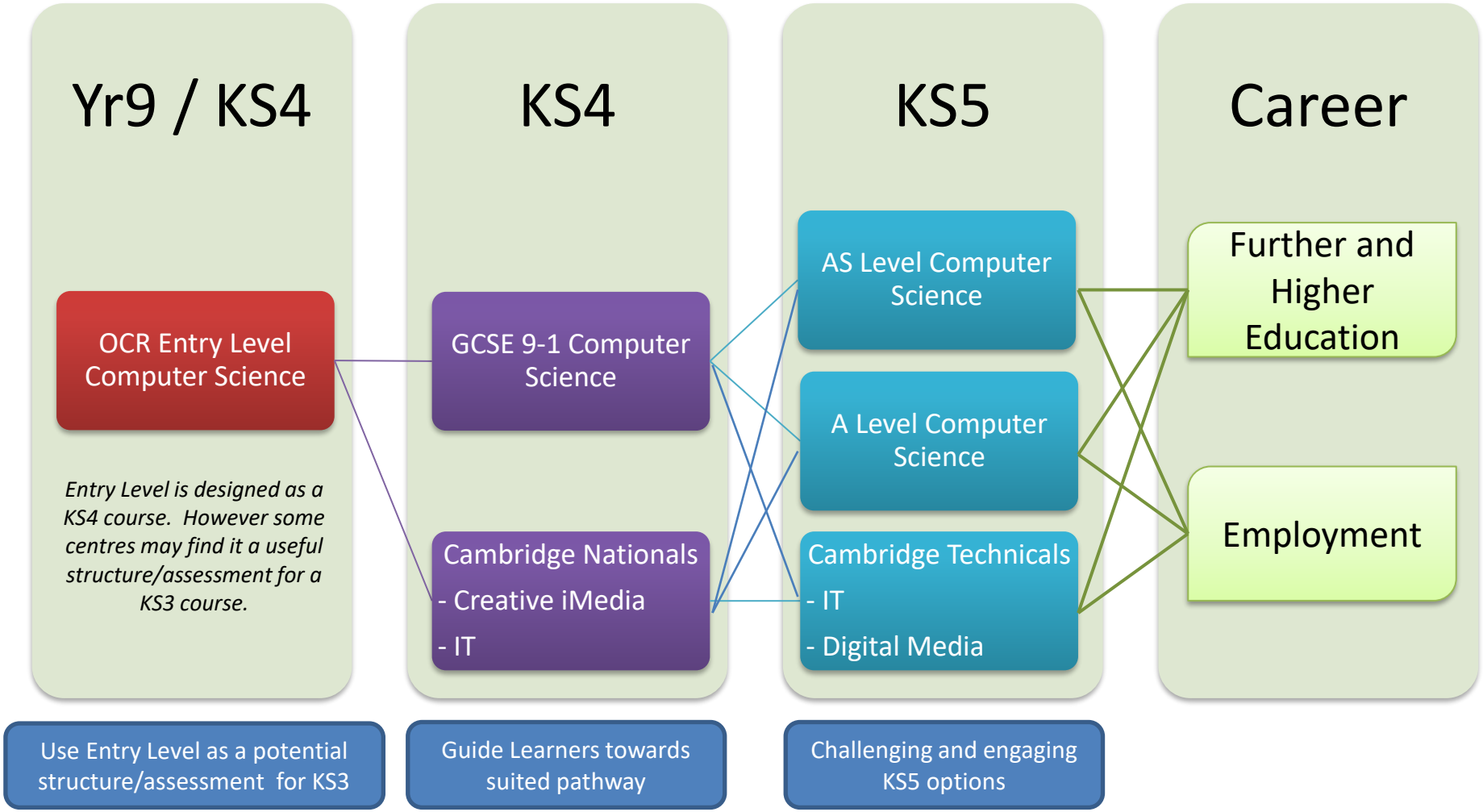
- An introduction
- Approaches to coding and learning
- Sharing ideas and practice for engaging students

Setting up Computer Science J277

Our teacher engagement and how it fed into development

OCR
Oxford Cambridge and RSA

Computing Qualifications Suite – Progression Path



OCR
Oxford Cambridge and RSA

Engagement with SLT

Considerations

- Computer Science is a science – this is important
- Scheme of Work flow from KS3 through to KS5
- Resources available:
  - Physical
  - Staffing
- Is this the right course?
- Do SLT understand the nature of Computer Science?
- Timetable hours allocation
- Baseline assessments?

Preparing with our Entry Level CS Course

- Our Entry Level is a **pre** L1/L2 course (i.e. pre-GCSE)
- Designed for learners who wouldn't usually reach L1 by 16
- However - can be used as a loose SoW for KS3
- Internally assessed
- Programming Project
- You can certify with OCR if you wish to – but no compulsion
- Use our GCSE resources – just remove some content/depth

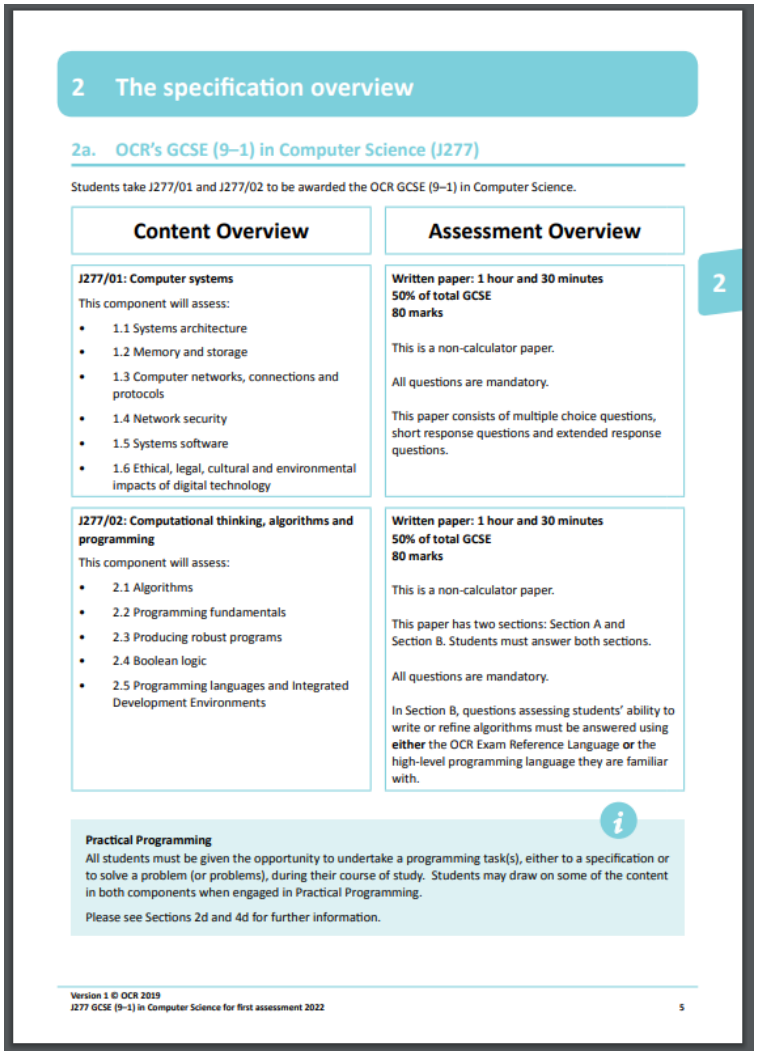
Diving in at the deep end

- You can just go straight in to CS at KS4
- However – can be a big step up for many candidates without prior knowledge
- If you do not have time at KS3 for whole CS offer, focus on problems solving and thinking skills
- We offer an Algorithms Challenge booklet which focuses on breaking down and expressing solutions to problems

Specification and assessment

OCR
Oxford Cambridge and RSA

Inside J277



2 The specification overview

2a. OCR's GCSE (9–1) in Computer Science (J277)

Students take J277/01 and J277/02 to be awarded the OCR GCSE (9–1) in Computer Science.

Content Overview	Assessment Overview
J277/01: Computer systems This component will assess: • 1.1 Systems architecture • 1.2 Memory and storage • 1.3 Computer networks, connections and protocols • 1.4 Network security • 1.5 Systems software • 1.6 Ethical, legal, cultural and environmental impacts of digital technology	**Written paper: 1 hour and 30 minutes** **50% of total GCSE** **80 marks** This is a non-calculator paper. All questions are mandatory. This paper consists of multiple choice questions, short response questions and extended response questions.
J277/02: Computational thinking, algorithms and programming This component will assess: • 2.1 Algorithms • 2.2 Programming fundamentals • 2.3 Producing robust programs • 2.4 Boolean logic • 2.5 Programming languages and Integrated Development Environments	**Written paper: 1 hour and 30 minutes** **50% of total GCSE** **80 marks** This is a non-calculator paper. This paper has two sections: Section A and Section B. Students must answer both sections. All questions are mandatory. In Section B, questions assessing students' ability to write or refine algorithms must be answered using **either** the OCR Exam Reference Language **or** the high-level programming language they are familiar with.

Practical Programming
All students must be given the opportunity to undertake a programming task(s), either to a specification or to solve a problem (or problems), during their course of study. Students may draw on some of the content in both components when engaged in Practical Programming.

Please see Sections 2d and 4d for further information.

- **Section 1:** Overview of why Computer Science matters and aims/learning outcomes
- **Section 2:** Overview of assessment model and topics covered: Help to shape your Long Term SoW
- **Section 3:** Understanding how our assessments work: 3b/c/d especially important
- **Section 4:** Administration requirements for GCSE

Section 2: Content

Sub topic	Guidance
1.2.3 Units	
☐ The units of data storage: ○ Bit ○ Nibble (4 bits) ○ Byte (8 bits) ○ Kilobyte (1,000 bytes or 1 KB) ○ Megabyte (1,000 KB) ○ Gigabyte (1,000 MB) ○ Terabyte (1,000 GB) ○ Petabyte (1,000 TB) ☐ How data needs to be converted into a binary format to be processed by a computer ☐ Data capacity and calculation of data capacity requirements	**Required** ✓ Why data must be stored in binary format ✓ Familiarity with data units and moving between each ✓ Calculate capacity of devices ✓ Calculate required capacity for a given set of files ✓ Calculate file sizes of sound, images and text files ▪ sound file size = sample rate x duration (s) x bit depth ▪ image file size = colour depth x image height (px) x image width (px) ▪ text file size = bits per character x number of characters **Alternatives** • Use of 1,024 for conversions and calculations would be acceptable • Allowance for metadata in calculations may be used

- Each topic contains multiple sub-topics
- Tick boxes to help you/students keep track of coverage
- Clear guidance on what you need to teach
- Alternatives/formulae/symbology included

Practical Programming Skills

2d. Practical Programming skills

All students must be given the opportunity to undertake a programming task or tasks during their course of study.

The programming task(s) must allow them to develop skills within the following areas when programming:

- Design
- Write
- Test
- Refine

Each task(s) must use one or more high-level text-based programming language, either to a specification or to solve a problem (or problems). They can use any high-level text-based programming language, such as:

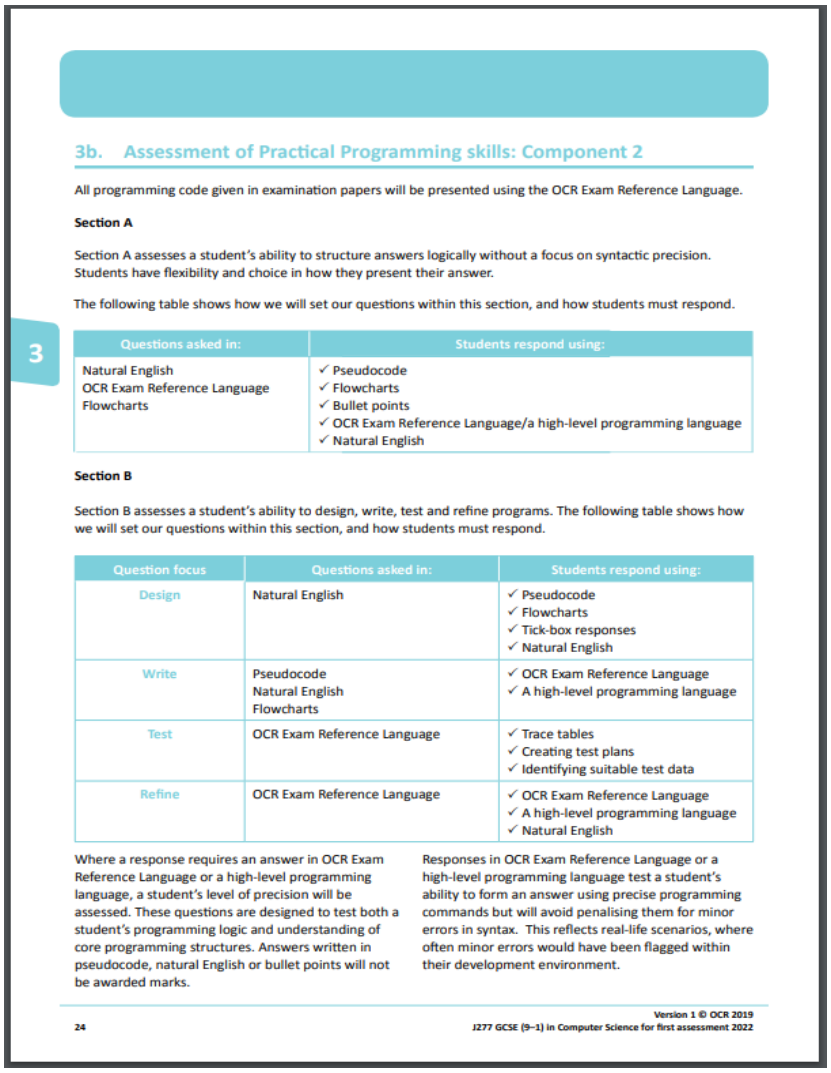
- Python
- C family of languages (C#, C++, etc.)
- Java
- JavaScript
- Visual Basic/.Net
- PHP
- Delphi
- BASIC

Some high-level languages do not allow demonstration of all the Practical Programming skills. Where this is the case, schools are encouraged to consider using a second language for practical experience.

Practical Programming skills will be assessed in Component 2 of the qualification, in particular Section B. See Section 3b 'Assessment of Practical Programming skills: Component 2' for more details.

- Requirements have **changed**
- Ensure you are up to date with the changes
- Core elements are now delivery of:
  - Design
  - Write
  - Test
  - Refine
- Centre must submit a practical programming statement

Section 3: Assessment



3b. Assessment of Practical Programming skills: Component 2

All programming code given in examination papers will be presented using the OCR Exam Reference Language.

Section A

Section A assesses a student's ability to structure answers logically without a focus on syntactic precision. Students have flexibility and choice in how they present their answer.

The following table shows how we will set our questions within this section, and how students must respond.

3

Questions asked in:	Students respond using:
Natural English OCR Exam Reference Language Flowcharts	✓ Pseudocode ✓ Flowcharts ✓ Bullet points ✓ OCR Exam Reference Language/a high-level programming language ✓ Natural English

Section B

Section B assesses a student's ability to design, write, test and refine programs. The following table shows how we will set our questions within this section, and how students must respond.

Question focus	Questions asked in:	Students respond using:
Design	Natural English	✓ Pseudocode ✓ Flowcharts ✓ Tick-box responses ✓ Natural English
Write	Pseudocode Natural English Flowcharts	✓ OCR Exam Reference Language ✓ A high-level programming language
Test	OCR Exam Reference Language	✓ Trace tables ✓ Creating test plans ✓ Identifying suitable test data
Refine	OCR Exam Reference Language	✓ OCR Exam Reference Language ✓ A high-level programming language ✓ Natural English

Where a response requires an answer in OCR Exam Reference Language or a high-level programming language, a student's level of precision will be assessed. These questions are designed to test both a student's programming logic and understanding of core programming structures. Answers written in pseudocode, natural English or bullet points will not be awarded marks.

Responses in OCR Exam Reference Language or a high-level programming language test a student's ability to form an answer using precise programming commands but will avoid penalising them for minor errors in syntax. This reflects real-life scenarios, where often minor errors would have been flagged within their development environment.

- **Section 3a:** Outlines how the exam papers are laid out.
  - Paper 1 remains the same
  - Paper 2 now has 2 sections
    - Section A - 50 marks
    - Section B - 30 marks
- **Section 3b:** Digs further into how the second paper will look and feel
- Note that some questions require a more **formal response** when writing algorithms

OCR
Oxford Cambridge and RSA

Section 3c: OCR Exam Reference Language

Concept	Keyword(s)/Symbols	Example
Iteration		

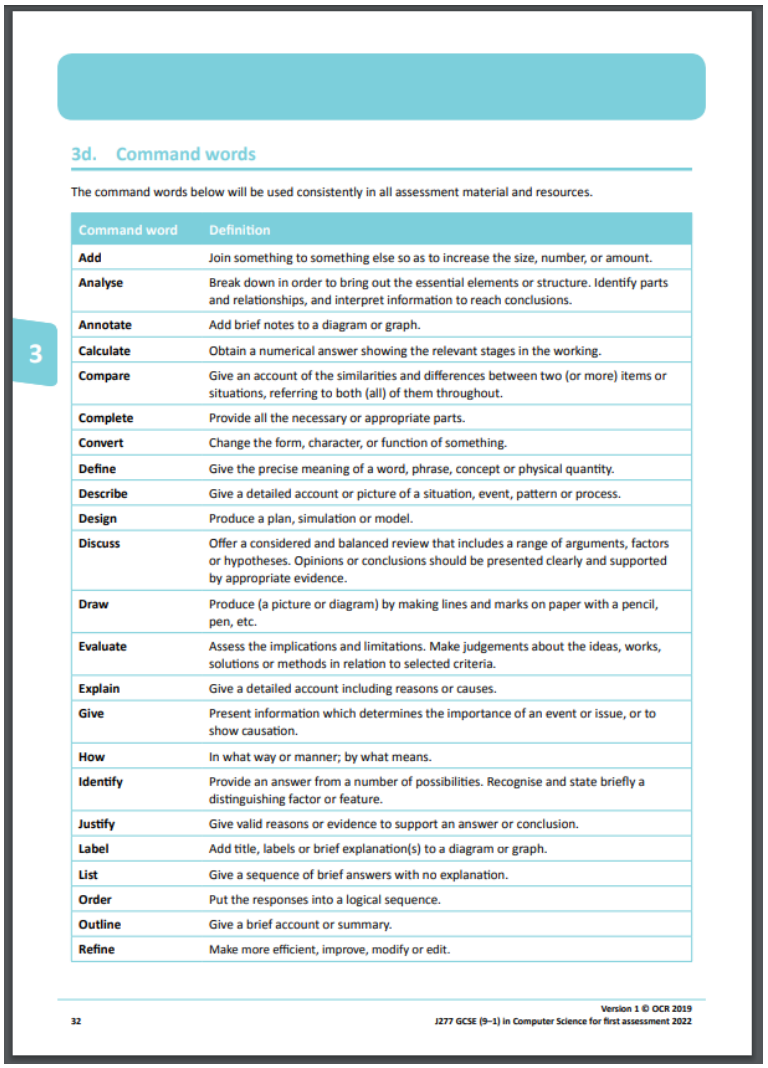
FOR loop
(Count-controlled)
	`for … to …`
	
	`next …`
	
	`for … to … step …`
	
	`next …`
		`for i=0 to 9`
		`    print("Loop")`
		`next i`
		This will print the word "Loop" 10 times, i.e. 0-9 **inclusive**.
		
		`for i=2 to 10 step 2`
		`    print(i)`
		`next i`
		This will print the even numbers from 2 to 10 inclusive.
		
		`for i=10 to 0 step -1`
		`    print(i)`
		`next i`
		This will print the numbers from 10 to 0 inclusive, i.e. 10, 9, 8,…, 2, 1, 0.
		
		Note that the `'step'` command can be used to increment or decrement the loop by any positive or negative integer value.

OCR Exam Reference Language is, for the majority, the same as J276 'Pseudocode Guide'. Familiarity and stability was key.

Section 3C: OCR Exam Reference Language

- We have exemplified each technique
- There have been some updates – ensure you have carried these forward in any resources you re-use!
- ERL may be used for any algorithm/programming response
- Where questions state specifically ERL or a high-level programming language response is required, more informal expressions such as pseudocode will not be credited

Section 3d: Command Words



3d. Command words

The command words below will be used consistently in all assessment material and resources.

Command word	Definition
Add	Join something to something else so as to increase the size, number, or amount.
Analyse	Break down in order to bring out the essential elements or structure. Identify parts and relationships, and interpret information to reach conclusions.
Annotate	Add brief notes to a diagram or graph.
Calculate	Obtain a numerical answer showing the relevant stages in the working.
Compare	Give an account of the similarities and differences between two (or more) items or situations, referring to both (all) of them throughout.
Complete	Provide all the necessary or appropriate parts.
Convert	Change the form, character, or function of something.
Define	Give the precise meaning of a word, phrase, concept or physical quantity.
Describe	Give a detailed account or picture of a situation, event, pattern or process.
Design	Produce a plan, simulation or model.
Discuss	Offer a considered and balanced review that includes a range of arguments, factors or hypotheses. Opinions or conclusions should be presented clearly and supported by appropriate evidence.
Draw	Produce (a picture or diagram) by making lines and marks on paper with a pencil, pen, etc.
Evaluate	Assess the implications and limitations. Make judgements about the ideas, works, solutions or methods in relation to selected criteria.
Explain	Give a detailed account including reasons or causes.
Give	Present information which determines the importance of an event or issue, or to show causation.
How	In what way or manner; by what means.
Identify	Provide an answer from a number of possibilities. Recognise and state briefly a distinguishing factor or feature.
Justify	Give valid reasons or evidence to support an answer or conclusion.
Label	Add title, labels or brief explanation(s) to a diagram or graph.
List	Give a sequence of brief answers with no explanation.
Order	Put the responses into a logical sequence.
Outline	Give a brief account or summary.
Refine	Make more efficient, improve, modify or edit.

- Command words are used consistently throughout the examination
- We list them here to help you use them accurately in assessments
- Also allows you to expose students to them and help them understand expectations

Refine command word has been added to the list

Section 3e: Assessment Objectives

3e. Assessment Objectives

There are three Assessment Objectives (AOs) in OCR GCSE (9–1) in Computer Science. These are detailed in the table below.

Students are expected to:

	Assessment Objective
AO1	Demonstrate knowledge and understanding of the key concepts and 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.

Assessment Objective weightings in OCR GCSE (9–1) in Computer Science

The relationship between the Assessment Objectives and the components are shown in the following table:

Component	% of overall GCSE (9–1) in Computer Science (J277)		
	AO1	AO2	AO3
Computer systems (J277/01)	21	29	0
Computational thinking, algorithms and programming (J277/02)	9	11	30
Total (%)	**30%**	**40%**	**30%**

- Three different assessment objectives (AOs)
- AOs weightings for Paper 1 and Paper 2
- AO3 will be examined only in Paper 2
- 1 public / 1 Interchange set of practice papers

Section 4d: Practical Programming Skills

4d. Practical Programming skills administration requirements

It is a requirement for all centres to complete and submit a Practical Programming Statement for each year in which students are entered for a GCSE (9–1) in Computer Science.

By signing the statement, your centre is confirming that it has given all students the opportunity to undertake a programming task or tasks during their course of study, as outlined in Section 2d.

The Practical Programming Statement can be downloaded from our website at www.ocr.org.uk

Any failure to submit to OCR a Practical Programming Statement in a timely manner may result in a malpractice/maladministration investigation.

Formal requirements for the **Practical Programming** skills have been updated.

Section 4d: Practical Programming Skills

- There are no time limits (minimum or maximum)
- There are no **required** tasks
- We provide suggestions
- You have **total** flexibility

Remember:

- You can teach the majority of Component 02 through programming
- This situates their learning in 'doing' – a powerful tool
- Practical work engages learners
- You can be very flexible in how you deliver this
- Whilst longer projects are good for combining skills...
- Shorter tasks help to 'cement' techniques into memory
- Try to be creative about where/how techniques can be used

  e.g. How many ways can you print 1 to 10

Teaching and Delivery

Keeping Computer Science Fun!

OCR
Oxford Cambridge and RSA

Resources for Delivery

- **Scheme of Work**
  - provides overview of the course, content updates/stability, links to resources, 2 potential delivery plans.
- **Content Mapping Guide**
  - provides broad oversight to the content changes from J276 to J277.
- **Exam Reference Language**
  - provides overview of the changes from J276 'pseudocode' to the J277 Exam Reference Language.
- **Programming Challenges**
  - provides short and longer challenges to help you delivery content, knowing that you will meet the programming skills required.
- **Teacher Delivery Packs**
  - provides comprehensive lesson plans, PowerPoints, quizzes and lesson activities as a base to plan from

Resources for Delivery

- **Sample Assessment Materials**
  - provides public facing examples of our assessments
- **Candidate Sample Answers**
  - provides commentary from the Principal Marks on candidate answers. These explain why certain responses may/may not have received credit.
- **Assessment Story**
  - provides a guide as to how we created our SAMs and provides good practice which you can use to build your own internal assessments.
- **End of Unit quizzes**
  - provide informal summative assessment ideas
- **Options Evening resources**
  - provides everything you need for options evening

Building Your Course Delivery

*Time allocation for topics will vary based on personal experience, cohort ability and progress. These timings are meant as a **guide only** and not a required or specified amount which is endorsed by OCR.*

Practical Programming forms an essential part of learning. We would encourage centres to use practical experience within the classroom to support lessons where possible.

Approximate teaching time	Topic	Sub Topic	Contained in J276	Modified from J276	New Content for J277	Resour
2	2.1.1 Computational thinking	☐ Principles of computational thinking				
		o Abstraction				
		o Decomposition				
		o Algorithmic Thinking.				
8	2.1.2 Designing, creating and refining algorithms	☐ Identify the inputs, processes, and outputs for a problem				
		☐ Structure diagrams				
		☐ Create, interpret, correct, complete, and refine algorithms using:				
		o Pseudocode				
		o Flowcharts				
		o Reference language/high-level programming language				
		☐ Identify common errors				
		☐ Trace tables				

1. Download the Scheme of Work
2. Read the Component 01 and Component 02 content
3. Map where curriculum content has changed (clearly shown in our Scheme of Work)
4. Review and modify one of the approaches or make your own!
5. Use the resources links to customise your delivery

Building Your Course Delivery

- Suggested timings are for guidelines only
- Links for resources are updated and modified from time to time
  - Check for new links or ideas!
- Think about how you can make lessons as practical and engaging as possible
  - e.g. Sorting and Searching using cards, or board games
- Use our Teacher Delivery Packs for starter ideas and build / differentiate from these

Practical Programming

Stays the same…

- As part of the course of learning, students are **required** to **design, write, test** and **refine** programs
- They must still use one or more high-level programming language with a textual program definition
- You can do ANYTHING you like as long as it meets these two criterion

You can now…

- Skills may focus on a single, or multiple tasks
- Centre submits an annual declaration of conformance
- Total freedom to pick tasks and activities for your students
- You can use Arduino, Micro:bit, Robotics, Machine Learning, online courses etc.

Remember

- There are small updates to programming skills content

Practical Programming

Stays the same…

- Lists of fundamental **skills** required shown in Specification section 2.1
- Algorithm fundamentals in section 2.1.2
- Lists of additional skills in section 2.3
- Freedom to choose languages you use

New Content

- Random Numbers
- Trace Tables
- Testing Plans
- Invalid Data
- Refining algorithms

We will continue to support you by providing a **range of programming ideas** to help engage and enthuse your students.

Using the Computer Science Community

- We are all learning and adapting
- Share resources at networks or Facebook
- Ask others - what they are doing?
- Steal ideas (and credit them!)
- Be adventurous

Let us know if you do something that works really well. We may ask you if you want to commission it as an OCR resource!

Third Party Resources

- We have a range of endorsed resources on the website
- There are many other 3rd party non-endorsed resources
- Always good to have a range and to mix and match
- Always cross reference resources with the Specification guidance

- Our endorsed publishers are all updating resources for J277
- Please speak to them directly for dates / release

Keeping up to date

Ensuring your knowledge is current!

OCR
Oxford Cambridge and RSA

Exam Results Analysis

You can use it to:

- Review reports on performance:
  - individual candidates
  - cohorts of students
  - whole centre

activeresults

https://www.ocr.org.uk/administration/support-and-tools/active-results/

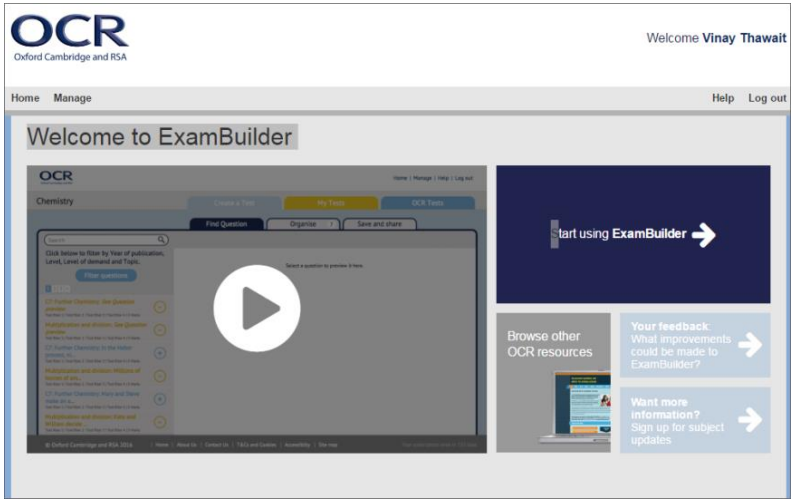
- Analyse results at question/topic level
- Compare your centre with OCR national averages or similar OCR centres
- It is available for all GCSEs, A Levels and Cambridge Nationals.
- Working on Cambridge Technicals availability

It allows you to:

- Identify trends across the centre
- Gain additional insight to support decisions e.g. whether to apply for an enquiry about results
- Facilitate effective planning and delivery of courses
- Identify areas of the curriculum where students excel or struggle
- Help pinpoint strengths and weaknesses of students and teaching

Exam Builder

- New FREE Mock Assessment Service
- Create Question Papers from past papers
- Tailor the questions or parts of questions, to closely reflect the topics you have covered
- Interchange account details required
- Past papers available to tailor the question, to closely reflect the topics you have covered
- Register now!



BECOME AN EXAMINER WITH OCR

GCSE and GCE Computer Science
Cambridge Nationals and Technicals

Examining with OCR can be a real benefit to your career, giving you;

- ✓ A better understanding of the specification.
- ✓ An insight in to how assessment decisions are made, helping to prepare your students effectively.
- ✓ Networking and knowledge sharing opportunities.
- ✓ Enhanced professional development.
- ✓ An additional boost to your income.

Examining is carried out online, allowing you to work flexibly around your busy schedule.

Full training and on going support are provided.

For further information: ocr.org.uk

Email: assessor.recruitment@ocr.org.uk *or call:* 01223 552 558

OCR
Oxford Cambridge and RSA

OCR
Oxford Cambridge and RSA

Stay Connected…

- **Join** the OCR J277 GCSE Computer Science Facebook group
  - https://www.facebook.com/groups/WeTeachJ277/

- **Twitter** (@OCR_ICT) – to find out about events and progress

- Sign up for email updates to ensure you are receiving the latest newsletters and informative emails
  - Our eNews letters will help you stay abreast of any developments and new resources etc.

- **Email** the team: ComputerScience@ocr.org.uk
- **Call** the team: 01223 553 998

Teacher Networks and Professional Development

- We have OCR specific "face-to-face" networks
- There are local CAS Communities to get support
- Both provide you with local support to help you
- We run Professional Development Courses to help you understand and develop knowledge of our specification and assessments.