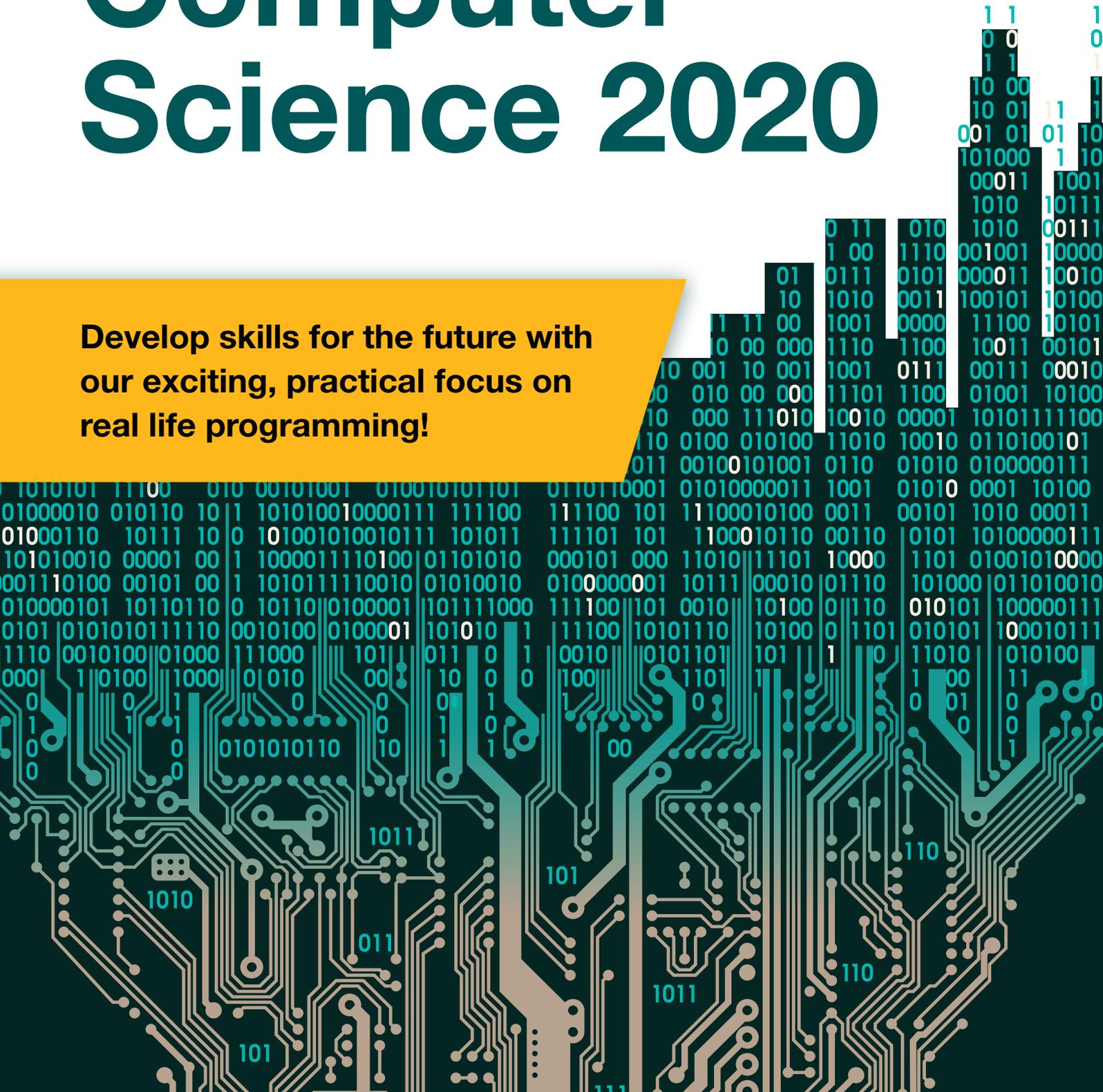


Pearson Edexcel  
GCSE (9–1)

# Computer Science 2020

Develop skills for the future with  
our exciting, practical focus on  
real life programming!



# Pearson Edexcel GCSE (9–1)

# Computer Science

This brochure will guide you through our Pearson Edexcel GCSE (9-1) Computer Science qualification from 2020. As both an awarding body and a publisher, we'll outline how our specification will inspire and engage your students, address changes to the qualification and also showcase the high quality support you can expect.

Our up-to-date qualification reflects the fast-changing world of Computer Science - with innovative, future looking onscreen practical assessments to ensure all students develop the computational skills they need for an exciting future beyond the classroom.

## Our qualification:

- 🔗 has up-to-date, practical on screen assessments, engaging and accessible for everyone .....**page 4**
- 🔗 features rich, future-facing, engaging content .....**page 6**
- 🔗 unbeatable and expert support every step of the way .....**page 8**

**Our Computer Science GCSE will see paper 2 taken onscreen, a practical onscreen assessment reflecting real-life programming in the modern world.**



# Teaching Computer Science for the future

**It's shaped by you, for your students.** That is why we've developed our GCSE Computer Science 2020 in collaboration with teachers and the computer science community, to create an engaging qualification that equips students with the knowledge and practical skills to thrive in the fast-changing world of Computer Science.

**The future of Computer Science is here.** Which is why, our qualification, the first of its kind, provides a practical approach to developing computational skills. This includes innovative, practical onscreen assessment to ensure all students develop the computational skills they need for an exciting digital future beyond the classroom.

“

*The practical onscreen assessment approach will prepare students well for eventual employment because it gives an opportunity for the learner to be assessed on writing, testing, and refining programs in an IDE using Python 3. Being given two hours to complete a task also relates well to the real life time constraints experienced by many Computer Scientists.*

Matt Hogan, Learning Manger (Secondary) at Raspberry Pi

”

“

*Can I thank/congratulate you on all producing three full sets of spec papers in addition to the public exemplar assessment questions - that is really fantastic effort and makes such a difference when teaching a new spec. The fact that we can engage with this level of feedback/discussion in advance of the first set of public assessment is testament to the superb resourcing that has gone into making this exciting practical launch a success!*

Paul Baker, Head of Computing & Digital Strategy, The Perse School

”



# Assessment Overview

## Paper 1 Principles of Computer Science

Paper code: 1CP2/01

✓ 75 marks

🕒 Written examination: 1 hour 30 minutes

⚖️ 50% of the qualification

### Content overview

This paper will assess Topics 1 to 5.

#### 🔗 Computational thinking

- understanding of what algorithms are, what they are used for and how they work; ability to follow, amend and write algorithms; ability to construct truth tables.



**Data** - understanding of binary, data representation, data storage and compression.



**Computers** - understanding of hardware and software components of computer systems and characteristics of programming languages.



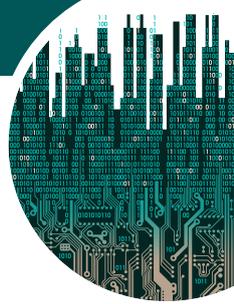
**Networks** - understanding of computer networks and network security.



**Issues and impact** - awareness of emerging trends in computing technologies, and the impact of computing on individuals, society and the environment, including ethical, legal and ownership issues.

### Assessment overview

This paper consists of five compulsory questions, each one focused on one of the topic areas. The questions consist of multiple-choice, short-, medium- and extended-open-response, tabular and diagrammatic items.



# Paper 2

## Application of Computational Thinking

Paper code: 1CP2/02

✓ **75 marks**

🕒 **Practical onscreen examination: 2 hours**

⚖️ **50% of the qualification**

### Content overview

This paper will assess Topic 6: Problem solving with programming.

#### The main focus of this paper is:

- ✓ understanding what algorithms are, what they are used for and how they work in relation to creating programs
- ✓ understanding how to decompose and analyse problems
- ✓ ability to read, write, refine and evaluate programs.

### Assessment overview

This paper is practical in nature and requires students to design, write, test and refine programs in order to solve problems.

Students will complete this assessment onscreen using their Integrated Development Environment (IDE) of choice.

They will be provided with:

- ✓ coding files
- ✓ a hard copy of the question paper
- ✓ Programming Language Subset (PLS) – as an insert in the question paper and an electronic version

Students should then answer the six compulsory questions onscreen using Python 3.

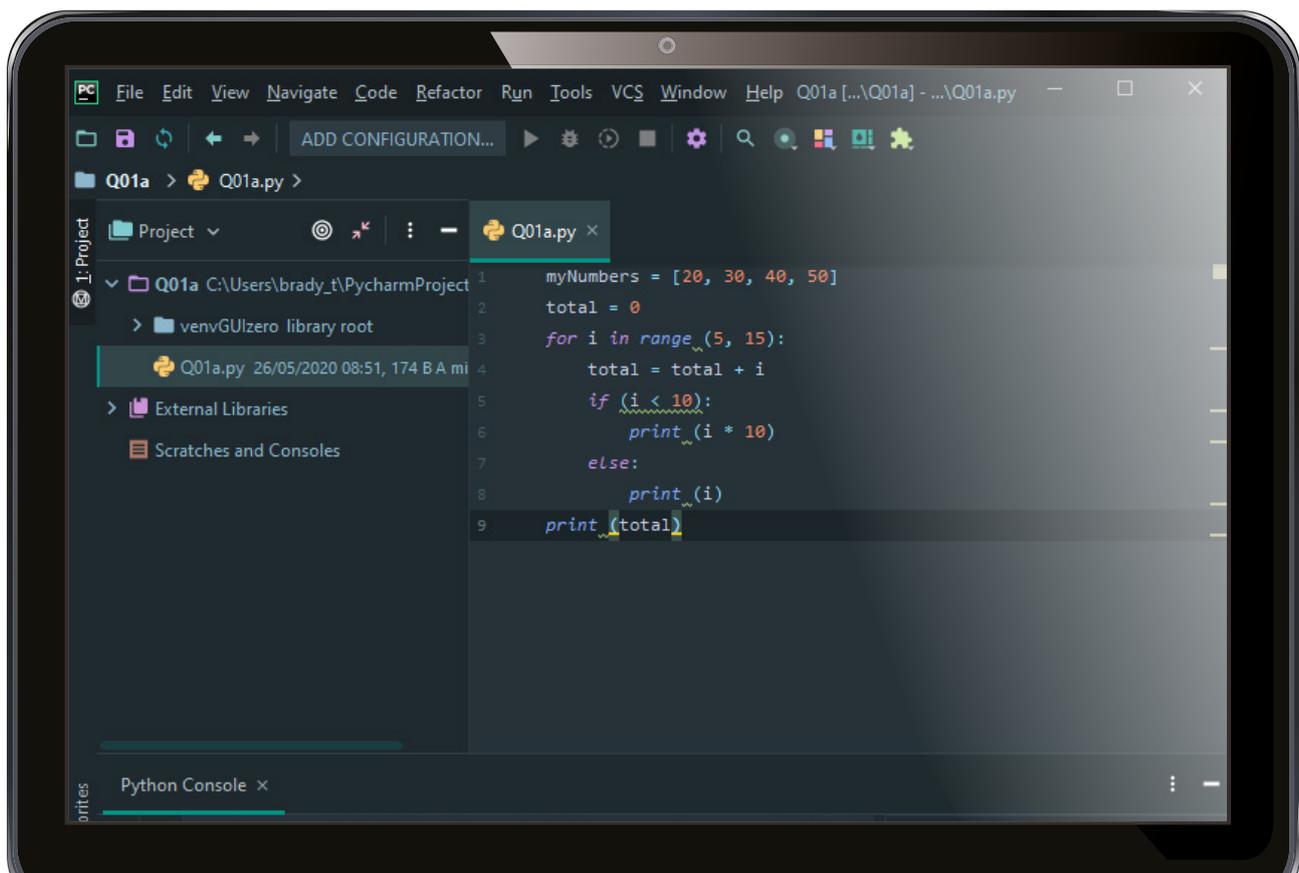


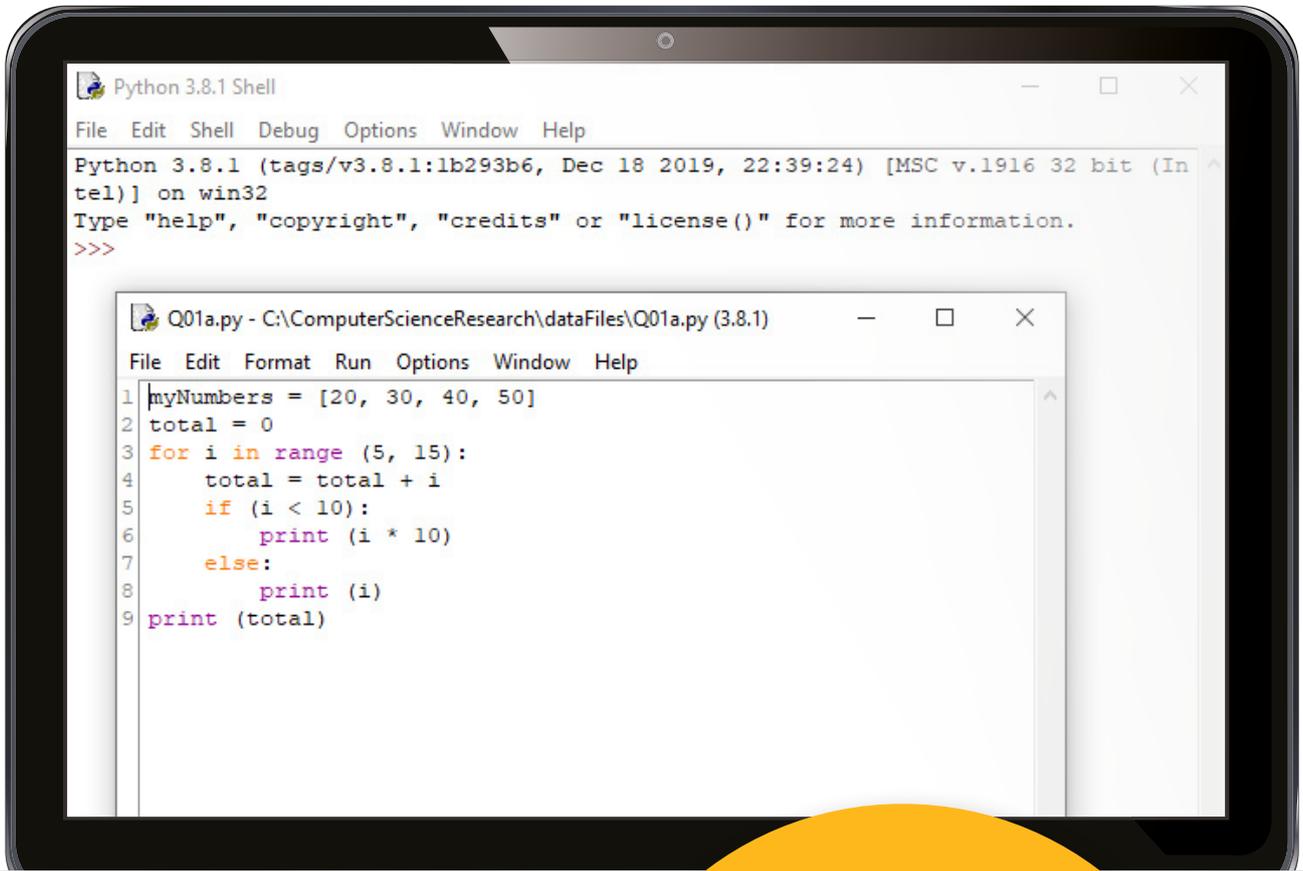
# Onscreen assessment

Our approach of assessing programming practically via our onscreen assessment allows schools to **choose which Integrated Development Environment (IDE) to use**. Students are then able to complete activities using their IDE of choice, so they're programming in a familiar environment - **without the need for internet access**.

Our assessments are designed with an **underpinning pedagogical approach** derived from NCCE Hubs, and ensure real-world programming skills are assessed via a realistic, practical and engaging assessment experience.

We've chosen **python** as a vehicle to facilitate the teaching, learning and assessment for all students to get the best possible experience.





Any questions  
about how this will work  
in practice?

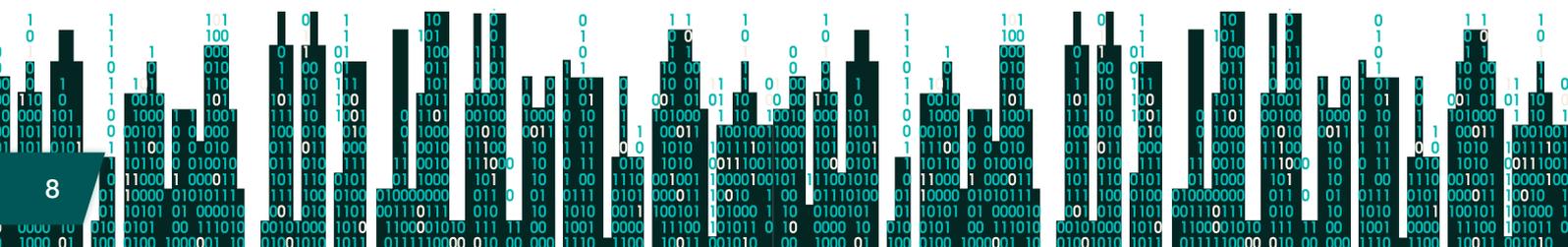
Please take a look at our **online  
FAQ document** or get in touch.

**quals.pearson.com/  
ComputerScienceGuide**



# Clear and simple structure

Our Pearson Edexcel GCSE (9-1) Computer Science qualification has a written and a practical element. This balance provides students with a rounded experience of computer science in comparison with a traditional paper based exam that relies on students reading and writing pseudocode. Our onscreen assessment will give them a more practical and engaging assessment experience that can be transferred into real world skills.





## Our qualification:

- ✔ **Has a straightforward structure.** There are six comprehensive topic areas, assessed through two externally-examined papers.

The **written paper** focuses on:

- computational thinking
- data
- computers
- networks
- the impact and issues of computing in the world today.

The **practical onscreen assessment** focuses on:

The ability to analyse and solve problems by designing, writing testing and refining programs in a real world setting.

- ✔ **Has equally weighted papers.** The balance of theory and practical assessment are equally weighted, with a total assessment time of 3hrs 30 mins and a total of 150 marks (75 marks per paper).
- ✔ **Has a reduced number of command words.** This will help students to better understand how they should respond to each question. We have identified specific command words for each paper and we won't stray from these.
- ✔ **Provides a clear and easily applied mark scheme** outlining exactly what students need to evidence.
- ✔ **Paper 2 doesn't require an internet connection.** The data files, Programming Language Subset (PLS) and the question paper can be downloaded on the morning of the exam. We have at least three channels prepped and ready to go to ensure that this works smoothly:
  - ✔ secure download from our website
  - ✔ secure download via Edexcel Online;
  - ✔ and, as a final resort the question paper delivery team are ready to secure file transfer on request.

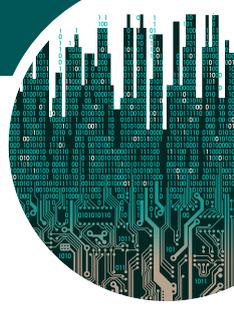


# Unbeatable and expert support every step of the way

On our website, you'll find a range of teaching and learning material to help you prepare for this qualification and assessment style, including:

- ✔ three sets of sample papers that will be ready long before the first live assessment, as we know how important it is that your students are prepared and familiar with the assessments before they sit their exam
- ✔ a Getting Started Guide, providing additional guidance that clarifies what needs to be covered for each topic area
- ✔ an interactive scheme of work
- ✔ lesson plans, activities and solutions
- ✔ a series of short videos that talk about the paper 2 assessment talking about the logistics of the exam and going through each question and mark scheme in details.
- ✔ Mocks Marking Training event.

**Turn over to discover more about our free support for planning, delivering, teaching and assessing Pearson Edexcel GCSE (9-1) Computer Science.**



Learn more about  
our partnerships

[quals.pearson.com/  
ComputerScienceGuide](https://quals.pearson.com/ComputerScienceGuide)

## The Interactive Scheme of Work

Specially designed resources aligned to the specification, bringing Computer Science to life with real-world examples, and providing support for the practical onscreen assessment.

Our **interactive scheme of work** (iSOW) supports the practical approach and assessment requirements of GCSE Computer Science and:

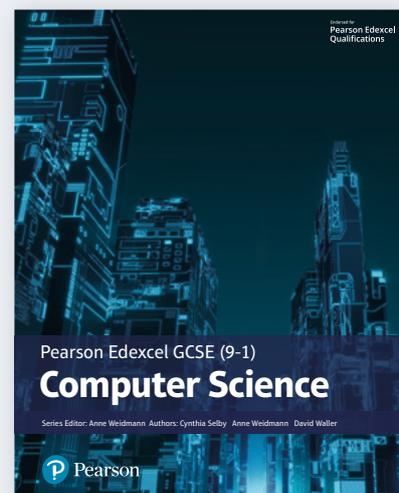
- ✔ recommends a **practical teaching route** and can be customised as much or as little as you want to support your level of **specialism**
- ✔ will save you time and effort in identifying the best resources, where best to use them in the classroom and how to get the most impact from them
- ✔ includes:
  - ✔ lesson plans, practical activities and solutions that have been written for the specification
  - ✔ a wealth of resources to support teaching Python that have been designed for you and for your students
  - ✔ specific links to high quality resources from leading pedagogical and industry organisations in Computer Science that have been chosen to support our qualification's practical approach including NCCE's resource PG Online, Microsoft's MakeCode, Craig'n'Dave's YouTube resources and many more.

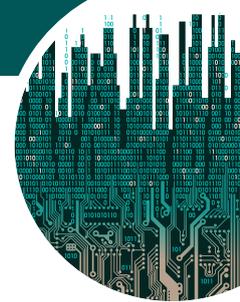
## The Student Book

Our **Pearson Edexcel GCSE (9–1) Computer Science Student Book** follows the practical teaching and learning approach of the interactive scheme of work and provides all of the content, activities and exam support that you need to deliver the specification.

- ✔ **An ActiveBook (eBook) version of the Student Book** is available for schools and colleges, with multiple user licenses to provide flexibility in the way you teach.

Take a look and download sample copies online  
[pearsonschools.co.uk/CSResources](https://pearsonschools.co.uk/CSResources)





Case studies of Computer Science in Action help to make the content more relevant and inspiring to all of your students.

Only the essential content has been included and the material has been simplified to make it as accessible as possible.

Packed with practical activities to support the practical approach of the Edexcel specification.

## Computational thinking

**Computer Science in action: Chess algorithms**

Algorithms for playing chess are used widely. After four moves by each opponent, there are over 288 billion possible further moves – far too many for a human to consider, but within the range of computers. This is what makes it possible for a top-level computer program to defeat a chess grandmaster.

**Key terms**

**Program code:** the implementation of an algorithm, in a human-readable form, that can be translated to a form that can be executed on a computer.

**Program code**  
In addition to flowcharts and written descriptions, algorithms can also be expressed in **program code**. Both flowcharts and written descriptions can be translated into programming languages.

There are steps to follow that will help produce programs that execute on a computer. Depending on the complexity of the original problem, some of these may be omitted, but for most problems, these steps show a sensible approach.

**Analyse** – understand the problem that needs to be solved and how to determine success.

**Design** – develop a solution to the problem and express the solution in a written description and/or a flowchart.

**Implement** – translate the description of the algorithm to a programming language.

**Debug and test** – using tools, execute, find errors, fix errors and test different inputs.

**Evaluate** – make judgements about the solution based on the original requirements of the problem and consider changes related to efficiency.

### Example of a simple algorithm

Here is an example of a simple algorithm. It sets two numbers, calculates the total by adding them together, and then displays the total on the screen. This algorithm does not need the user to type in any numbers. Three different versions of this algorithm are shown: a written description; a flowchart; and program code.

### Written description

**Worked example**

Below is an algorithm for adding two numbers.

Set the first number to 11.

Set the second number to 22.

Calculate total by adding first and second numbers.

Output total.

26

## Computational thinking

**PRIMM Activity**

This activity uses the code shown here. Use it for each of the steps shown.

```

1 #
2 # Global variables
3 #
4 b = 22
5 h = 44
6 a = 0
7
8 #
9 # Main program
10 #
11 a = (1/2) * b * h
12 print ("Area is", a)

```

**Predict:**  
What do you think the code will do? What output do you think it will produce?

**Run:**  
Load the code into your coding environment and run it. Did it do what you thought it would do? Did the output match your prediction? If not, how did it differ?

**Investigate:**  
Remove the line 'b = 22'. Run the code. Carefully read the error message. What is the message trying to tell you? Put the line back in. Change the calculation to move the '(1/2)' between the 'b' and 'h'. Run the code. What does that tell you about the use of brackets? What does that tell you about the order of arithmetic operators?

**Modify:**  
Be sure to run the code after each change to check it still works. Change the identifier names (variables) to be more meaningful. Remember to use camel case, if required. Using the existing output line, add lines to display the values of each variable before the total area.

**Make:**  
A program is needed to calculate the price of an item after adding tax. The program must store the base price of an item, the tax rate (expressed as a decimal), and the final price. The program must output the base price, the tax rate, and the final price. The formula to calculate the final price is base price x (1 + tax rate).

30

Filled with worked examples and activities using the PLS or 'Programming Language Subset' (Python) for a simple and comprehensive approach to programming.

Packed with exam questions using only the Edexcel command words so that exam preparation is concise and productive.

Every section in Part A (Topics 1 and 6) includes a practical activity for programming that follows the PRIMM pedagogy.

\*Pearson's paid for resources, as well as other endorsed resources, are not a prerequisite for the delivery of Pearson Edexcel specifications.

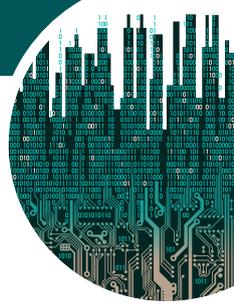


# Events

When it comes to tracking progress and preparing for assessment, we'll provide support and resources to help you and your students throughout the course.

- ✔ Planning and Delivering the new GCSE Computer Science specification from 2020.
- ✔ Pearson Edexcel GCSE (9-1) Computer Science: Introduction to Onscreen Assessment followed by training for each question on paper 2 (6 additional recordings).
- ✔ Pearson Edexcel GCSE (9-1) Computer Science: Introduction to Onscreen Assessment Specimen Set 1 followed by training for each question on paper 2 (6 additional recordings).
- ✔ Introduction to Onscreen Assessment Specimen Set 2 followed by training for each question on paper 2 (6 additional recordings).
- ✔ Pearson Edexcel GCSE (9-1) Computer Science: Introduction to Onscreen Assessment Specimen (Set 3) followed by training for each question on paper 2 (6 additional recordings).
- ✔ Feed back on the summer series
- ✔ Mocks marking training
- ✔ New event coming in January - Preparing for the Summer Exam

Find out more about our published resources at:  
[quals.pearson.com/CSRecording](https://quals.pearson.com/CSRecording)



# Assessment & Progression

When it comes to tracking progress and preparing for assessment, we'll provide support and resources to help you and your students throughout the course.

- ✔ **Sample assessment materials**, so that you can get to grips with the format of the papers and the level of demand as quickly as possible.
- ✔ **Exemplar materials with commentaries** for both examined papers.

## Tools to help you track progress

### ResultsPlus

ResultsPlus provides the most detailed analysis available of your students' exam performance. Widely used by teachers across the country, this free online service enables you to identify topics and skills where students could benefit from further learning, helping them gain a deeper understanding of computer science.

[quals.pearson.com/resultsplusCS](https://quals.pearson.com/resultsplusCS)

Paper 2 is further supported by the provision of fully granular 'Item level mark data' ILMD for every student in your cohort. The additional detail in the ILMD will enable you to perform very detailed forensic analysis of your candidates individual performance in Paper 2.

### examWizard

examWizard is a free exam preparation tool containing a bank of past Pearson Edexcel GCSE Computer Science exam questions, mark schemes and examiners' reports.

[quals.pearson.com/examwizardCS](https://quals.pearson.com/examwizardCS)



# Your Subject Advisor

If you need any support or advice, feel free to get in touch with our dedicated subject advisor.

## Tim Brady

Tim is an ex-IT teacher himself and can answer your questions about the Pearson Edexcel GCSE (9-1) Computer Science 2020 qualification.



 **0333 016 4160**

 **[Quals.pearson.com/contactusCS](https://quals.pearson.com/contactusCS)**

## Get in touch

### Pearson Edexcel GCSE (9-1) Computer Science 2020

For queries, information and support, we're happy to help.

 Email us: **[TeachingComputerScience@pearson.com](mailto:TeachingComputerScience@pearson.com)**

 **Book a 1-2-1 session with Tim**

 Visit us online: **[quals.pearson.com/ComputerScienceGuide](https://quals.pearson.com/ComputerScienceGuide)**

Find out more about our published resources at:  
**[pearsonschools.co.uk/CSResources](https://pearsonschools.co.uk/CSResources)**

