



Ada Lovelace(1815-1852)

“The Analytical Engine has no pretensions whatever to originate anything. It can do whatever we know how to order it to perform. It can follow analysis; but it has no power of anticipating any analytical relations or truths. Its province is to assist us to making available what we are already acquainted with.

[Describing Charles Babbage's machine.]”

Contents

OCR Computer Science - H446	3
Content overview	3
Assessment overview	4
Preparing for the course.....	5
Exploring A Level Topics	5
Strengthening your foundations of the GCSE course.....	5
Progress check to determine suitability of course	5
Bridging work.....	6
Required Tasks for Bridging Work Submission	7
Task 1	7
Task 2	8
Task 3	9
Submission Instructions	10
Course Textbook.....	11
Want to extend your learning?	12
Podcasts	12
Books	13
Movies.....	14

OCR Computer Science - H446

Welcome to Computer Science at A level. We are delighted you have chosen to study Computer Science at Newstead Wood School and look forward to meeting you in September. In this booklet you will find:

- Course Content
- Assessment details
- Bridging Work
- Wider learning resources

To prepare you for programming, we ask that you complete the Bridging Work over the summer and bring this to your first lesson in September. **You will be assessed on your understanding of this Bridging Work during the induction assessment week in September.**

Content overview

Component 01: Computer systems

Students are introduced to the internal workings of the (CPU), data exchange, software development, data types and legal and ethical issues. The resulting knowledge and understanding will underpin their work in component 03.

It covers:

- The characteristics of contemporary processors, input, output and storage devices
- Types of software and the different methodologies used to develop software
- Data exchange between different systems
- Data types, data structures and algorithms
- Legal, moral, cultural and ethical issues.

Component 02: Algorithms and programming

This builds on component 01 to include computational thinking and problem-solving. It covers:

- What is meant by computational thinking (thinking abstractly, thinking ahead, thinking procedurally etc.)
- Problem solving and programming – how computers and programs can be used to solve problems
- Algorithms and how they can be used to describe and solve problems.

Component 03: Programming project

Students are expected to apply the principles of computational thinking to a practical coding programming project. They will analyse, design, develop, test, evaluate and document a program written in a suitable programming language. The project is designed to be independently chosen by the student and provides them with the flexibility to investigate projects within the diverse field of computer science. We support a wide and diverse range of languages.

Assessment overview

Students must take all three components to be awarded the OCR A Level in Computer Science.

Computer systems (01)

Duration: 02 hours 30 mins Marks 140 Weightage : 40%

Content: The internal workings of the (CPU), data exchange, software development, data types and legal and ethical issues.

Calculators not allowed.

Algorithms and programming (02)

Duration: 02 hours 30 mins Marks 140 Weightage : 40%

Content: Using computational thinking to solve problems.

Calculators not allowed.

Programming project (03)

Marks 70 Weightage 20% Non-exam assessment.

Students will be expected to analyse a problem (10 marks), and design (15 marks), develop and test (25 marks), and evaluate and document (20 marks) a program. The program must be to solve it written in a suitable programming language.

Preparing for the course

As you prepare to begin A Level Computer Science, it is worth reflecting on your confidence across the GCSE Computer Science curriculum, particularly in programming. A Level builds on many of the concepts studied at GCSE but explores them in significantly greater depth and complexity.

Exploring A Level Topics

You may find it useful to look over the specification and topic links here [OCR A Level Computer Science H446 Specification](#) to identify the common areas. Craig and Dave provide a full range of videos that cover the full specification. <https://craigndave.org/ocr-a-level-h046-h446-videos/>

Isaac Computer Science is a free online resource that gives you access to a huge range of online learning materials. Create your free login account and refer to the A level OCR content to prepare:

<https://isaacomputerscience.org>

Tips to prepare: <https://isaacomputerscience.org/pages/top-tips-a-level>

Strengthening your foundations of the GCSE course

To succeed in Year 12 Computer Science, it is essential to have a strong foundation in the GCSE curriculum before you begin in September. Refreshing the GCSE content over the summer will ensure you have the necessary knowledge and skills to fully engage with the more advanced topics covered in Year 12. This preparation will give you a head start, boost your confidence, and help you keep up with the fast pace of the course. Without this solid groundwork, you may find it challenging to grasp key concepts, which could affect your progress and overall performance. Therefore, dedicating time to thoroughly review and complete the GCSE curriculum during the holidays is a crucial step towards achieving success in your A-level studies.

Craig and Dave provide a full range of videos that cover the full GCSE specification.

<https://craigndave.org/ocr-gcse-j277-videos/>

Isaacs Computer Science is a free online resource that gives you access to a huge range of online learning materials for the classroom, homework, and revision. Create your free login account and refer to GCSE OCR content to prepare.

<https://isaacomputerscience.org>

Progress check to determine suitability of course

During the induction assessment week in September, all students enrolled in the Computer Science course will be assessed on their understanding of the GCSE curriculum. A minimum standard of performance will be expected in order to continue with the course. Students who do not meet this threshold may be advised to consider an alternative subject option.

Bridging work

To help prepare you for Computer Science in Year 12 we would like every student to experience Python Programming.

If you haven't studied Python before start by exploring the following websites:

www.w3schools.com

<https://time2code.today/python-course>

If you want a refresher, please enrol in the free course available on the FutureLearn website. You will have four weeks to complete it from the date you register, so be sure to choose a time when you can commit to finishing it within that period. This also covers Object Orientation which will be useful for your A level course.

[Introduction to Programming with Python - Online Software Development Course](#)

Once secure try and attempt the challenges provided on

Isaac Computer Website:

https://isaacomputerscience.org/gcse_programming_challenges

Required Tasks for Bridging Work Submission

Find below the three Tasks you are expected to submit as part of your bridging work.

Task 1

<https://www.ocr.org.uk/images/260930-coding-challenges-booklet.pdf> **(Complete any two of the OCR Challenge Problems numbered 20 or above.)**

Select and complete any two challenge problems from the OCR challenge set, **choosing from Challenge Problem 20** onwards from document above that best reflect your current level of ability. For each one, submit your **annotated code in Python and the screenshot of the output**. I personally use IDLE to code and run Python, which is free to download and install. As an alternative to IDLE you can always use online python editors. I also use “trinket” which will work on smart phones, with no downloads needed.

Your work will be assessed based on:

- **Level of complexity** demonstrated in your solution
- **Clarity and purpose of annotations**
- **Efficiency and structure** of your code or algorithm
- **Use of appropriate programming constructs**
- **Testing and evidence of accuracy (validations or outputs)**
- **Readability and formatting**

Optional suggestion: You may include a brief reflection on the approach you took and any challenges you encountered.

Tip for annotation: Anyone can see what the code does — a good annotation explains the *reason* behind it.

- *Bad:* # Add 1 to x
- *Good:* # Increment x to move to the next index in the list

Task 2

The Cornell method of note taking:

The expectation to do independent research at A Level will increase dramatically from GCSE. There is a real skill to taking decent notes outside of lesson which are of value.

Research has proven that one of the most effective methods is the “Cornell” note taking method.

1. To start download the “Cornell note taking template” from Craig n Dave:
 - a. [Cornell note-taking - Craig 'n' Dave](#)
2. Pick any two of the following videos from Craig ‘n’ Dave:
 - a. [OCR A Level \(H406-H466\) SLR1 – 1.1 Fetch, decode, execute cycle - Craig 'n' Dave](#)
 - b. [25. OCR A Level \(H446\) SLR5 – 1.2 Stages of compilation - Craig 'n' Dave](#)
 - c. [27. OCR A Level \(H046-H446\) SLR6 – 1.2 Development methodologies part 1 - Craig 'n' Dave](#)
 - d. [32. OCR A Level \(H446\) SLR7 – 1.2 Programming paradigms - Craig 'n' Dave](#)
 - e. [48. OCR A Level \(H446\) SLR9 – 1.3 Symmetric & asymmetric encryption - Craig 'n' Dave](#)
 - f. [68. OCR A Level \(H446\) SLR12 – 1.3 PageRank algorithm - Craig 'n' Dave](#)
3. Write the title of the video and its topic in the top boxes (use a different sheet for each video).
4. In the main “Notes” section, write notes from the video. You can do this in any way you like, a suggestion might be to rewind slightly when the canvas changes, thinking carefully about what was important in the previous few minutes.
5. Having recorded the notes, review them:
 - Turn each part into a question in the section on the left.
 - For example, the notes may say, “The value of the program counter is passed to the memory address register”.
 - The question then becomes, “Which register is the value of the program counter passed to?”
 - Sometimes these questions are easy, and at times they are more difficult to write.
 - There may also be more than one valid question.
 - You will need to decide for yourself which are the most appropriate questions for revision.
 - Finally pull out all the key words and their definitions words the notes and list them in the bottom section

Task 3

Part 1: Personal Reflection (Approx. 200 words)

Question:

Why have you chosen to study A-Level Computer Science?

Consider the following in your response:

What interests you about the subject?

How do you think Computer Science will help you in the future?

Any specific career goals or passions related to computing?

What previous experience (if any) do you have with programming or technology?

Part 2: Research Task – Explore a Computer Science Topic

Choose one topic from the list below to research over the summer.

Write a summary (300-400 words) covering:

- What the topic is about
- Why it's important in Computer Science
- How it might be used in real life or industry
- Any interesting facts or recent developments related to it

Topic options:

- Artificial Intelligence and Machine Learning
- Cybersecurity and Ethical Hacking
- Software Development Life Cycle (SDLC)
- Data Structures and Algorithms
- The Impact of Computing on Society and Privacy

Reference:

Include any sources you used for your research.

Purpose of this task:

This task helps me understand your motivation and interests, as well as your research and writing skills. It will also give you an early insight into some key Computer Science topics before the course starts.

Submission Instructions

All tasks must be submitted via email to me (jmehta@newsteadwood.co.uk) by 1st September. Please combine **Tasks 1, 2 and 3** into a **single Word document**. Clearly label each section as Task 1, Task 2 and Task 3 within the document.

Name your file using the following format:

Y12BWFirstnameSurname

Task 1:

State the coding challenge number and the programming challenge you have chosen. Include screenshots of your annotated code as well as screenshots of the output screen where applicable. At the end of the screenshots, provide a brief review of your code.

Additionally, please include a link to your code. You may use any of the following platforms to share your code link:

<https://replit.com>

<https://github.com>

<https://pastebin.com>

Task 2

Two cornell style note sheets with notes on two videos of your choice

Task 3

Submit your reflection and research summary as a single task .

Course Textbook

The PG Online textbook provides comprehensive coverage of the course content we deliver. I've included a link to the PG Online website for your reference, which includes the book's details and ISBN. You're welcome to purchase it from any retailer of your choice—many offer competitive prices. While it's not essential at this stage, obtaining a copy before the course begins and familiarising yourself with the topics covered over the two years would be beneficial.

OCR AS and A Level Computer Science

Textbook PM Heathcote and RSU Heathcote ISBN: 978-1-910523-05-6

<https://www.pgonline.co.uk/resources/computer-science/a-level-ocr/ocr-a-level-textbook>

Want to extend your learning?

For a broader understanding of computer science and related fields, you might find the following resources helpful:

Podcasts

Programming throwdown: The show covers diverse topics in programming, offering insights into the latest tech trends, tools, and techniques.

<https://www.programmingthrowdown.com/>

Software Engineering Radio: Aimed at those passionate about software engineering, this podcast offers deep insights into the latest technological advancements. With over 450 episodes, it serves as a key resource for professionals and students eager to understand the fast-paced evolution of the tech industry.

<https://se-radio.net/>

Twit - This Week in Tech: Hosted by Leo Laporte, this show features a roundtable discussion on high-tech trends, with a strong emphasis on key tech companies and innovations. It keeps listeners up to date on advancements in areas such as AI, cloud computing, and emerging technologies.

<https://twit.tv/>

Hard Fork (The New York Times): Journalists Kevin Roose and Casey Newton make sense of the fast-changing world of technology, with a big focus on AI, ethics and how tech is reshaping our lives and jobs. Smart, balanced and easy to follow — a great way to think about the bigger questions behind computing and where it's all heading.

[Hard Fork - The New York Times](#)

Books

Computer Science: An Overview by J. Glenn Brookshear; Pearson, 2014. An introduction to computer science, presenting each topic with its history, current state, and future potential, including ethical considerations. It's a comprehensive guide for students and professionals alike.

Code: The Hidden Language of Computer Hardware and Software by Charles Petzold; Microsoft Press, 2000. This book connects everyday objects like flashlights and seesaws to computers, demonstrating how human ingenuity has driven technological innovation. It's a must-read for anyone curious about the fundamentals of computing.

The Pattern on the Stone: The Simple Ideas That Make Computers Work by Daniel Hillis; Basic Books, 1999. It explains basic computer concepts in accessible language. This book simplifies the complexities of computer science for readers of all backgrounds.

The Information: A History, a Theory, a Flood by James Gleick; Fourth Estate, 2012. A chronicle of how information has become the defining quality of the modern era. Gleick's narrative intertwines scientific discoveries with their historical context.

Outnumbered: From Facebook and Google to fake news and filter-bubbles – the algorithms that control our lives by David Sumpter; Bloomsbury Sigma, 2018. An applied mathematician explores how algorithms use our data and affect our lives. Sumpter provides a critical analysis of our algorithm-driven world.

Once Upon an Algorithm: How Stories Explain Computing by Martin Erwig; MIT Press, 2017. This book explains computer science concepts using familiar stories like Hansel and Gretel, Sherlock Holmes, Groundhog Day, and Harry Potter. Erwig's storytelling approach makes complex ideas more relatable.

Artificial Intelligence: A Ladybird Expert Book by Michael Wooldridge; Michael Joseph Books, 2018. Written by our Head of Department, it traces the evolution of intelligent machines from Turing's ideas to modern digital assistants like Siri and Alexa. Wooldridge provides a succinct overview of AI's history and its current state.

Computational Fairy Tales: Jeremy Kubica; CreateSpace, 2012. This book offers a whimsical journey through computational thinking, illustrating core computer science concepts through fairy tales. It's a creative introduction for young students to the world of computing.

Movies

The Imitation Game: This film is a true story during World War II. A British mathematician who is one of the pioneers in modern computers, Alan Turing, tries to decrypt the message contained in the Enigma machine. Alan Turing has been named the father of cryptography and the father of modern computers.

You can learn about cryptography, plain text, and ciphertext from this film. Besides that, what is also important is learning to understand what a Turing test is, a non-deterministic Turing machine and a deterministic Turing machine.

Hidden Figures : String link to computer science through its depiction of African-American women mathematicians and engineers who made significant contributions to NASA during the Space Race. The story highlights the work of Katherine Johnson, Dorothy Vaughan, and Mary Jackson, whose expertise in mathematics and computing was crucial for space missions. Their efforts in developing and implementing early computer programs and algorithms showcase the essential role of computer science in advancing technology and achieving monumental scientific milestones.

The Social Dilemma: This documentary on Netflix illustrates the profound impact of social media algorithms on human behaviour and society. It links to the computer science curriculum by showcasing real-world applications of data science, machine learning, and ethical computing, highlighting the importance of understanding algorithm design and its societal implications.

The Social Network: This film tells us about Mark Zuckerberg's journey in building Facebook. Not only that, there are many other sides of Mark Zuckerberg that we can observe in this film, including how he conflicts with the person who stole his data and with his co-workers. The film is set in Harvard.

Snowden: This film is a true story about Edward Snowden's journey as a worker at the National Security Agency. He realised that the United States government was spying on everyone through electronic devices to track and identify threats to the country. Snowden leaked this secret to the public and made him the most wanted fugitive.

We look forward to welcoming you on the course in September.

Wish you a lovely summer break!

Mrs Mehta

Head of Computer Science