

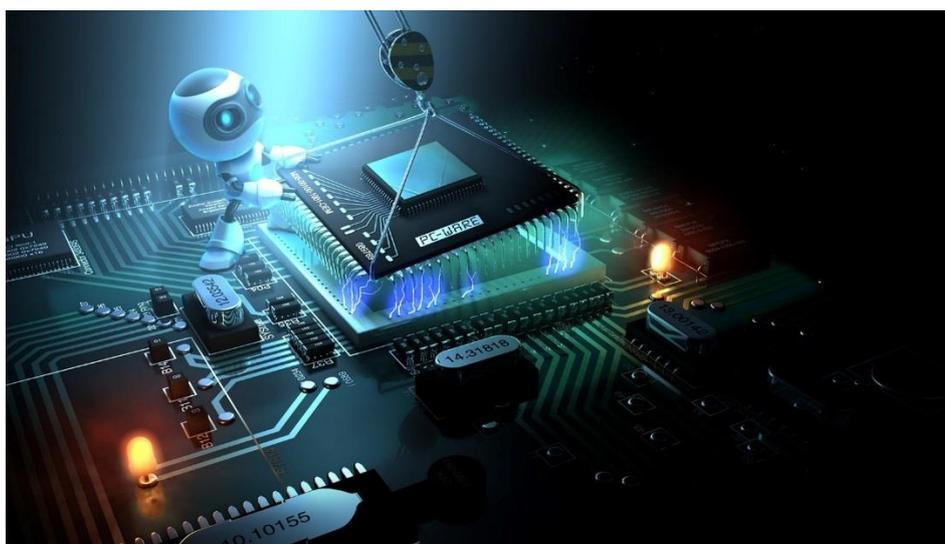


Beths Transition Pack for A-level Computer Science

Get ready for your A-Levels!

This guide is designed to help you get ready for the A-level
Computer Science course at Beths Grammar School.

This pack includes everything from the course content,
programming & theory based activities, suggested reading, online
documentaries and tutorials.



The transition from working at a GCSE standard to an A-Level is significant, including an increasing emphasis on technical content, extended answers and independent research. This pack is designed to allow you to practice some of these skills, building on the work that you may have covered at GCSE. Whether you have studied GCSE Computer Science or not, and whatever your grade, there will be something here to support your preparation for A-Level.

This transition pack is organised into the following sections:

- Course Overview
- Activities and Useful Links
- Computer Science Theory
- Algorithmic Thinking and Problem Solving
- Writing and Practising Code
- Enrichment Activities

This broadly matches the examination and non-examination assessments of the new GCSEs and A-Levels. Within each section there will be practice questions to support both the content and style of writing required at A-Level, plus various links to books and other resources that you can use to study any topics that require attention. Each section is based on the GCSE specification, so that the content should be familiar if you have already studied GCSE Computer Science; if you are new to the subject, this should give you an overview of the main topic areas that you will study.

The questions are designed to go beyond GCSE standard and prepare you for A-Level study. Some questions are quite straightforward, and test core knowledge. Others are chosen to give you a chance to extend both your thinking and writing skills and to demonstrate your creativity in solving problems. There are also some genuinely hard extension questions if you want them!

There are many different ways that you can use this resource, including:

- As a baseline assessment of skills before starting a course
- To support the development of specific skills
- As a resource to support bridging courses or other pre-course study

Note that this is not a “self-study” document on its own. This resources contains questions, prompts, starting points and solutions to help you study one or more core topics before starting the A-Level. However, it is not a text book and you may need to refer to the support resources, freely available online resources or books in order to complete the exercises. A number of interesting books for purchase are also listed. They are simply books that other staff, or students, have recommended. I leave you to evaluate them yourself.

OCR A Level Computer Science

Welcome to A Level computer science. Over the next two years we will heavily upgrade some of your GCSE skills and learn about totally new and cutting-edge topics such as artificial intelligence and data science. While studying the **OCR A Level (Course Code: H446)** you will prepare for two examinations.

Exam 1: Computer Systems 40% 2 hours and 30 minutes written paper (140 marks)

Computer systems component (01) contains the majority of the content of the specification and is assessed in a written paper recalling knowledge and understanding of the topics below.

- The characteristics of contemporary processors, input, output and storage devices
- Software and software development
- Exchanging data
- Data types, data structures and algorithms
- Legal, moral, cultural and ethical issues

Exam 2: Algorithms and Programming 40% 2 hours and 30 minutes written paper (140 marks)

Algorithms and programming component (02) relates principally to problem solving skills needed by learners to apply the knowledge and understanding encountered in Component 01.

- Elements of computational thinking
- Problem solving and programming
- Algorithms to solve problems and standard algorithms

The Project: 20% 70 marks

There is also a non-examined element to consider (20%). This is a substantial project based on a topic that you will select. It might be something that you have taken further interest in as part of the course or outside of it. You are not restricted to Python in terms of the programming for this project. This is a fantastic opportunity to develop an idea over time and document your successes (and failures, as they are useful too) and to see something through from start to finish under the scrutiny of yourself and others.

Recommended Text Book



OCR AS and A Level Computer Science

Author: PG Online Ltd

ISBN: 978-1-910523-05-6

Publisher: PG Online Ltd

Date: September 2016

Activities to get you started.

Recommended Books

- PG Online OCR AS and A Level Computer Science by PM Heathcote
- Computational Fairy Tales by Jeremy Kubica.
- Artificial Intelligence: A Ladybird Expert Book by Michael Wooldridge.
- The Cuckoo's Egg: Tracking a Spy Through the Maze of Computer Espionage by Cliff Stoll.
- The Code Book: The Secret History of Codes and Code-breaking by Simon Singh

Online Lectures / Documentaries

[BBC Click](#) - Your guide to all the latest gadgets, websites, games and computer industry news

<https://www.wired.co.uk/topic/technology> - The latest tech news from WIRED on giants like Apple and Google, and the most important technology issues of the day

<https://student.craigndave.org/a-level-videos> - An amazing bank of videos covering practically all of the A level course content.

<https://www.physicsandmathstutor.com/computer-science-revision/a-level-ocr/> - Tonnes of revision notes, flashcards, videos and past paper questions.

Free Online Courses

AI/Robotics, Game Development, Coding, Cyber Security and lots more free courses.

<https://www.futurelearn.com/subjects/it-and-computer-science-courses>

A short “crash courses” on Computer Science topics

<https://www.youtube.com/watch?v=tplctygH29Q&list=PL8dPuualjXtNIUrzyH5r6jN9ullgZBpdo&index=1>

Here you will find a tutorial on how to do just about anything you need to do with Python. It's fantastic and comes with exercises so that you can check it's all sinking in!

https://www.w3schools.com/python/python_syntax.asp

When you want a challenge that requires some problem solving, try this. They all come with working solutions, but they might not always be the same as yours as this is the beauty of programming!

<https://www.practicepython.org/>

The British Informatics Olympiad. Lots of hard coding challenges. Like the maths challenge, only for programming!

The Mayan Calendar is a good starting point.

<https://www.olympiad.org.uk/problems.html>

Computer Science Theory

Recommended resources

- OCR GCSE textbooks
- PG Online OCR AS and A Level Computer Science by PM Heathcote

Wider computing issues and integrated questions

These questions require you to use your technical knowledge in context. Reference any sources that you use to help you.

1. Create a timeline showing the history of computing, including any key discoveries or inventions. Extend your timeline to show how you think computer science might develop over the next 50 years.
2. Compare the Xbox ONE, PS4 Pro and PC as gaming platforms. You must use as much technical detail as possible and reference any evidence presented. Choose how you will present your ideas.
3. Discuss the benefits and limitations of Virtual Reality
 - a. In business contexts, such as medicine
 - b. As a gaming tool
 - c. As an extension to social media
4. Design the next piece of wearable technology, annotating how it will function and explain the function and purpose of any components used.

Systems Architecture

1. Produce an annotated diagram showing how the CPU processes data. This should include

Videos

Everyone can Code

A video from Apple demonstrating that coding is open to a wide range of people.

<https://www.apple.com/education/everyone-can-code/>

Anybody Can Learn Code:

Will.i.am and friends talk about why to get into coding and also give a small insight into the working environment.

<https://www.youtube.com/watch?v=nKlu9yen5nc>

- a. The purpose of the CPU
- b. Common CPU components and their function

- i. Arithmetic and Logic Unit (ALU)
 - ii. Control Unit (CU)
 - iii. Cache
 - iv. Registers
 1. Memory Address Register (MAR)
 2. Memory Data Register (MDR)
 3. Program Counter
 4. Accumulator
 - c. Reference to the fetch-execute cycle
2. Discuss, with examples, how the performance of a CPU can be improved, including:
 - a. Increasing the clock speed
 - b. Increasing the cache size
 - c. Increasing the number of processing cores

Memory

1. Compare RAM and ROM
2. Explain the need for virtual memory in a computer system
3. Describe the characteristics of flash memory

Storage

1. Complete the following table comparing optical, magnetic and solid state storage media

	Capacity	Speed	Portability	Durability	Reliability	Cost
Optical						
Magnetic						
Solid State						

2. Justify one use of each storage method

Networks

1. Explain the similarities and differences between
 - a. A LAN and a WAN
 - b. Client-server and peer-to-peer networks
2. Explain the difference between the Internet and the World Wide Web

3. Describe the factors that affect network performance, and explain how network performance can be improved
4. Draw three different network topologies
 - a. Label all the components required to create each network
 - b. Explain the purpose of each component in the network, including
 - i. Wireless Access Points
 - ii. Routers
 - iii. Switches
 - iv. Network Interface Cards
 - v. Transmission media, such as Ethernet Cables
5. Create an interactive presentation or resource, such as a website, that explains how your computer connects to a webpage, such as www.bbc.co.uk Include the following:
 - a. DNS
 - b. Hosting
 - c. TCP/IP, including the concept of layers
 - d. HTTP/HTTPS
 - e. Packet switching
6. There have been many recent high-profile cyber-attacks across the world, including the attack on the NHS in May 2017. Some commentators have said that “we now rely too much on technology”. Write an essay explaining how far you agree with this statement and including descriptions of threats to IT systems and ways to reduce vulnerabilities.

Systems Software

1. Create a presentation comparing Windows, Linux, iOS, Android (which is based on Linux) and Unix. Discuss the features of each operating system, comparing the benefits and limitations of each. Note that you can try a basic Unix interface here: <http://www.masswerk.at/jsuix/>

Ethical, Legal, Cultural and Environmental Concerns

Find a recent news story on one of the following topics:

- A legal issue in computing, such as a breach of the Data Protection Act
- An ethical issue in computing, such as the development of AI
- An environmental issue in computing, such as the disposal of waste equipment
- A technical development in computer science, such as the Internet of Things

Summarise the story, explaining any technical content for a student in year 10.

Explain how the story affects you as a student of computer science.

Computational Thinking – Theory

Computational Logic and Calculations

1. Complete the truth tables for the following expressions

a. A AND (B OR C)

A	B	C	B OR C	A AND (B OR C)
0	0			
0	0			
0	1			
0	1			
1	0			
1	0			
1	1			
1	1			

b. (NOT A) OR (NOT B)

i. What single logic gate produces the same result as this expression?

A	B			

c. Draw a circuit to represent each expression

2. Calculate each of the following, showing any appropriate working you need

- a. 13 MOD 2
- b. 16 MOD 6
- c. 15 MOD 3
- d. 7 MOD 8
- e. 13 DIV 2
- f. 16 DIV 6
- g. 15 DIV 3
- h. 7 DIV 8
- i. 2^0
- j. 2^7
- k. 2^8
- l. 2^{10}

3. Convert the following into the units given
 - a. 4 bytes = bits
 - b. 1 TB = bytes
 - c. 80 kB = GB
 - d. 40 MB = nibbles

4. Complete the table, converting between binary, hexadecimal and denary as required

Binary	Hex	Denary
0010 1010		
	0B	
		255
0110 0111		
	F5	
		48
	CD	

5. Complete the following calculations
 - a. 0110 0011 + 0011 0001
 - b. 1010 0110 + 1100 1111
 - c. 0110 0011 << 2 (bit shift left two places)
6. Check if these are valid ASCII characters. If they are, give their character equivalent. Note that the first bit is a check digit using even parity, and the remaining 7 bits are the character
 - a. 1110 0010
 - b. 1100 0111
 - c. 0011 0110
 - d. 1100 1010

Programming Tools and Standards

1. Compare the use of jpg, png and gif to store images, explaining the benefits, properties and uses of each image format
2. Produce an annotated diagram of the IDE you prefer to use to write code, explaining any features of the IDE that help you to produce your code. You may need to show examples of the IDE in use to highlight the different features

Algorithmic Thinking and Problem Solving

The following puzzles will help you to develop your logical thinking skills. There are many good books of puzzles, plus countless online sources to test your skills. Some recommendations are given later.

The following puzzles are representative of classical problems and problem solving strategies. You can solve each one by trial and error, but you are encouraged to think about the strategy you employed to solve the problem. Note that there are discussions of each problem available online if you want to investigate them in more detail.

Two good general strategies to try are:

- Can you solve a simpler version of the problem first?
- Can you draw a diagram to help you *visualise* the problem?

After that, you have your standard computer science strategies:

- Decomposition
 - Can you split the problem down into smaller parts to solve?
- Abstraction
 - Can you remove any unnecessary details to focus in on only what you need to solve the problem?
 - Be careful – are you sure that you have kept the right information?
- Generalisation and problem recognition
 - Is this puzzle a specific example of a problem for which there is a general solution? If so, how does it apply in this case?
 - Do you recognise the problem from somewhere else, or is it similar to something else?
 - You may need to generalise the problem to identify the core features so that you can spot equivalent problems.

Another important strategy is to ensure that the problem is *well-defined*. This means that you know:

- The goal: what you are trying to achieve
- The givens: what you know at the start, or your starting conditions
- The resources: what you have available to solve the problem
- The constraints: any rules that limit your solution
- The ownership: who or what is carrying out each part of the solution

Sometimes just working through the problem definition carefully is enough to give the required insight.

Programming Practice & Challenges

It would be fantastic if you were developing your programming and problem-solving skills over the summer. Below are a few useful resources that

1. Here you will find a tutorial on how to do just about anything you need to do with Python. It's fantastic and comes with exercises so that you can check it's all sinking in!

https://www.w3schools.com/python/python_syntax.asp

2. When you want a challenge that requires some problem solving, try this. They all come with working solutions, but they might not always be the same as yours but as long as yours works, that's what counts.

<https://www.practicepython.org/>

Challenge 1 - How many days have you been alive?

DON'T Google ANYTHING UNTIL TASK 6!

Task 1: Decomposition of the initial problem:

Task 2: What are three (at least) considerations you need to make when thinking about calculating how many days you have been alive?

Calculation

Calculate the number of days **you** have been alive on paper

Check your answer https://www.1happybirthday.com/birthday_calculator.php

Task 3: Generalisation

Adjust your method so that anybody could work out how many days **they** have been alive?

Draw a flow chart of your method.

Task 4: Abstraction

Consider the differences between a person using your method and a computer. Explain the differences between what a human would need compared to what a computer would need to accomplish the task at hand.

Task 5: Application

Write the days alive program using (<https://repl.it/>) or (<https://trinket.io/features/python3>)

It should, allow anybody to enter their birth date and correctly calculate the number of days they have been alive. You can compare your answer to the previous website.

Task 6: Consideration

How useable is your program? Is it robust, can you make it crash? Is it visual or text based only? How could you improve your application to make it more appealing? Try to complete some of the changes you have considered here to improve the experience of using your program.

Task 7: Research

Is there a different way to accomplish this programming task?

If you find out a different way, try to use it in your own program.

Explain the differences between the two methods.

Which method is more appropriate?

Coding Challenge 2 – Customer ID Generator

- a. Ask the user to input
 - i. Their first name
 - ii. Their surname
 - iii. A date, in the format DD/MM/YYYY
- b. The program should then output a customer ID as follows:
 - i. The date in the format YYYYMMDD, then the first three letters of the surname, then the first initial, then the length of their first name. All letters should be in capitals
 - ii. For example, John Smith, 27/05/2017 would give 20170527SMITHJ4
- c. The program should validate any inputs and keep asking for inputs until the user enters correct details or types “quit” at any point

Plan your algorithm first, using a flowchart or pseudocode

Code your algorithm, and provide evidence of both your code and the working output

Create a test plan for your algorithm, including testing your validation with normal, boundary and erroneous data

Coding Challenge 3 – Product Calculator

- d. Ask the user to input
 - i. The name of a product
 - ii. Its cost in pounds
 - iii. The program should keep asking for inputs until the user types “None”
- e. The program should then output:
 - i. The name and price of the most expensive item
 - ii. The name and price of the least expensive item
 - iii. The average price of the items
 - iv. The total cost of the items
 1. Items over £50 get a 5% discount
 2. VAT is added at the end at 20%
- f. The program should validate any inputs

Plan your algorithm first, using a flowchart or pseudocode

Code your algorithm, and provide evidence of both your code and the working output

Create a test plan for your algorithm, including testing your validation with normal, boundary and erroneous data.

Enrichment Activities

National Museum of Computing, Bletchley Park (Near Milton Keynes)

<http://www.tnmoc.org/>

<https://www.bletchleypark.org.uk/>

<http://www.codesandciphers.org.uk/bletchleypark/> (virtual tour)

The National Museum of Computing and the Bletchley Park code breaking exhibition are both on the same site, although each has a separate entrance fee. Huge range of technology to explore, including Colossus, the world's first electronic computer.

Museum of Science and Industry, Manchester

<http://msimanchester.org.uk/>

The museum has an exhibition covering the development of computers, and they have "Baby" the world's first stored program computer. (There is an interactive talk about Baby every day.)

Science Museum, London

<http://www.sciencemuseum.org.uk/>

A wide range of science and technology exhibitions. In particular, the museum is currently hosting an exhibition on robotics, charting our 500 year quest to make machines human.

Centre for Computing History, Cambridge

<http://www.computinghistory.org.uk/>

A large collection of vintage and retro computers, with an emphasis on how computers have developed over time and the social context and impact of technological change.