



Computer Science Transition workbook

- The topic of **Computer Science** is at the heart of the modern world
- Studying it can make you extremely sought after in today's job market
- The transition from GCSE to A level is significant, this includes:
 - An increased emphasis on **technical content**
 - An increased emphasis **independent research**

This workbook is designed to allow you to practice some of these skills and build on your existing knowledge. Each slide has a number in the top left to indicate the exercise number. Ensure you keep your work safe.

Please complete by your first lesson back in September. Bring to your first lesson.

A close-up, slightly blurred photograph of computer hardware, showing a network switch or patch panel with several yellow Ethernet cables plugged in. The background is dark, and the focus is on the cables and the ports.

The course is assessed by 2 exams
(40% each exam)
&
1 NEA (programming project (20%))

Emerging computer technology

Expected time to complete: 2 hours

In this task you get to investigate any area of emerging computer technology which interests you.

You can pick any area which interests you, but examples could be:

- Artificial intelligence
- Robotics
- Automated self driving cars
- Quantum computing

In no more than ONE side of A4 summarise the area you have chosen under the following four headings:

1. What is it?
2. What are the possible Social, Moral, Cultural and Ethical **benefits** of this technology on society
3. What are the possible Social, Moral, Cultural and Ethical **risks** of this technology on society
4. My conclusion on this technology and what it will mean for our world 10 years from now

Additional help:

For additional help and support in structuring your answer you might like to watch some of the videos from the following Craig 'n' Dave playlists:

OCR:

SLR 17 – Ethical, morale and cultural issues

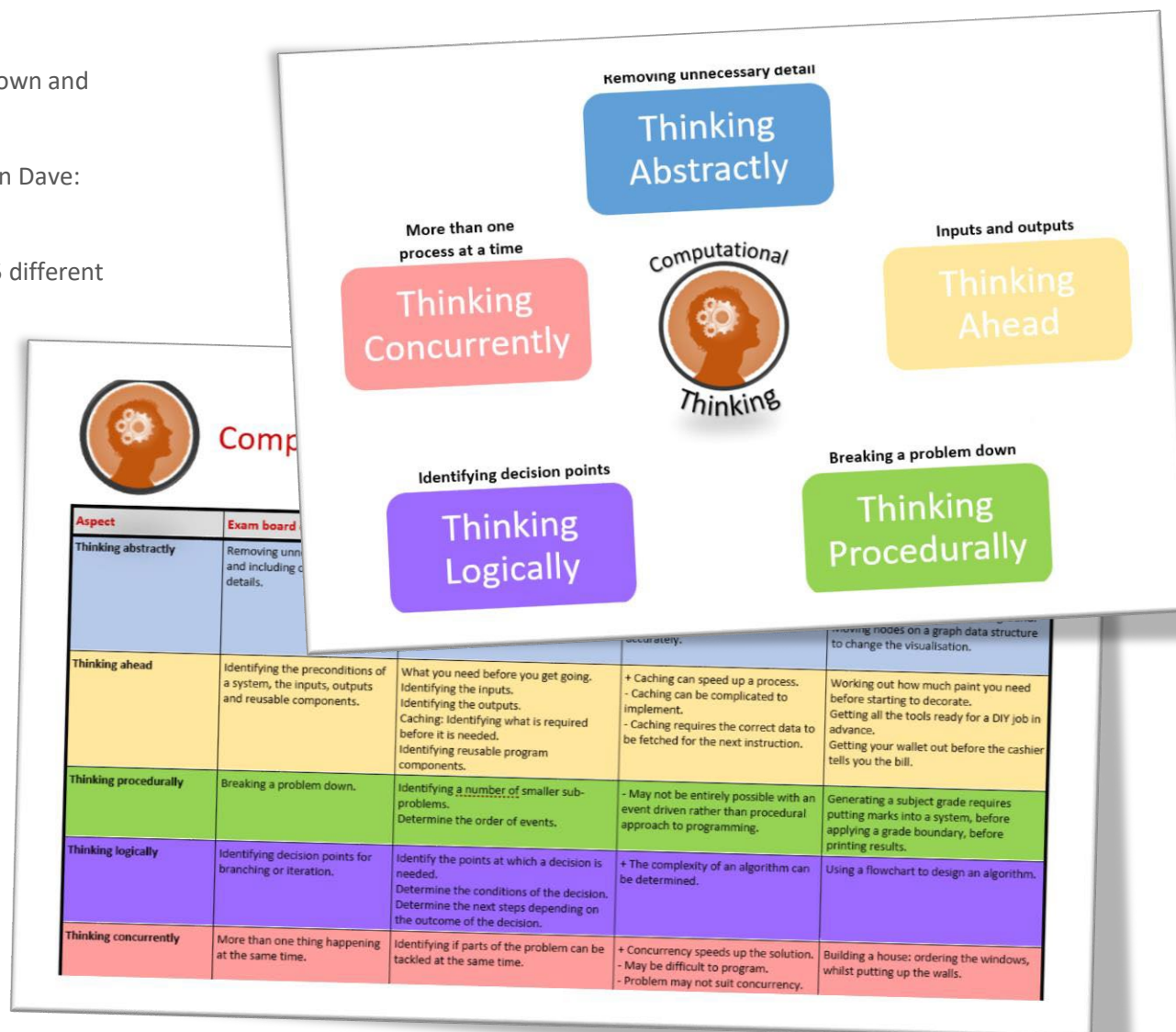
<https://www.youtube.com/watch?v=4h5zIBOgZz0&list=PLCiOXwirraUBrBvmXk-cwEYki7YOOkCgjC#>

Thinking like a computer

Expected time to complete: 2 hours

At the heart of Computer Science is the ability to look at problems, analyse them, break them down and solve them in a way which involves a variety of "Computational Thinking" skills.

- Download the "Computational thinking and Computational methods placemats" from Craig n Dave:
 - <https://student.craigndave.org/specification-key-terminology-and-cheat-sheets>
- Create your own spider diagram / mind map which shows your clear understanding of the 5 different computational thinking strands
 - Keep it to a single side of A4 / A3
- Your goal is to imagine someone else has to revise from your mind map. Ask yourself:
 - Does it make sense?
 - Is it clear?
 - Does it cover all of the important concepts?



The spider diagram illustrates five strands of computational thinking around a central 'Thinking' icon:

- Thinking Abstractly** (Blue): Removing unnecessary detail
- Thinking Concurrently** (Red): More than one process at a time
- Thinking Ahead** (Yellow): Inputs and outputs
- Thinking Logically** (Purple): Identifying decision points
- Thinking Procedurally** (Green): Breaking a problem down

The table below provides details for each strand:

Aspect	Exam board
Thinking abstractly	Removing unnecessary detail and including only the most important details.
Thinking ahead	Identifying the preconditions of a system, the inputs, outputs and reusable components. What you need before you get going. Identifying the inputs. Identifying the outputs. Caching: Identifying what is required before it is needed. Identifying reusable program components.
Thinking procedurally	Breaking a problem down. Identifying a number of smaller sub-problems. Determine the order of events.
Thinking logically	Identifying decision points for branching or iteration. Identify the points at which a decision is needed. Determine the conditions of the decision. Determine the next steps depending on the outcome of the decision.
Thinking concurrently	More than one thing happening at the same time. Identifying if parts of the problem can be tackled at the same time.

In A-Level Computer Science we will be studying a variety of programming languages other than just Python.

You can use one of the following languages but this list is not exhaustive:

C family of languages (for example C# C+ etc.)

Java

Unity game engine

Visual Basic PHP

Delphi

JavaScript

Python

Use YouTube video tutorials to start learning, there are plenty of online tutorials for all of these languages

HINT: You will probably first need to download and install an IDE for your chosen language I would recommend using Visual studio code.

Getting to grips with terminology

An important aspect of being successful with your study of Computer Science is getting to grips with subject related terminology. There are over 240 specific terms you will need to learn!

Below are a handful of the key terms you will need to become familiar with.

Control Unit	Register	Busses
Von Neuman Architecture	Optical Storage	Operating System
Intermediate Code	Device Driver	Compiler
Assembly Language	Machine Code	Lossy Compression
Hashing	Normalisation	TCP/IP Stack
Packet Switching	ASCII	Problem Decomposition

1. Research each of the key terms and write a definition.
2. Resist the urge to simply cut and paste a definition from the first website you find. Many definitions found on The Internet are overly complicated and wordy.
3. Ask yourself:
 - Does my definition make sense?
 - Is it succinct, to the point?
 - Does the definition have appropriate depth and detail for A'Level?
 - Could I give this definition to another student so they could revise from it?

Expected time to complete: 2 hours

