

Saracens High School



SARACENS
HIGH SCHOOL

A-Level Computer Science Transition booklet

Name: _____

Tell me about yourself

Why did you choose Computer Science?

In this simple task you get the opportunity to tell me your choices and reasons behind choosing to study Computer Science. Please answer all questions as best you can.

1. Why did you choose to study A-Level Computer Science?

.....
.....

2. What other courses have you chosen to study at Key Stage 5, and what made you choose this combination?

.....
.....
.....

3. What are you hoping to achieve from studying Computer Science?

.....
.....
.....

4. How would you describe yourself as a learner at GCSE? What skills were you good at, what areas would you like to improve on?

.....
.....
.....

5. What are your other hobbies and interests outside of school? Anything relating to Computing?

.....
.....
.....

Independent research task

Emerging computer technology

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 include:

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

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

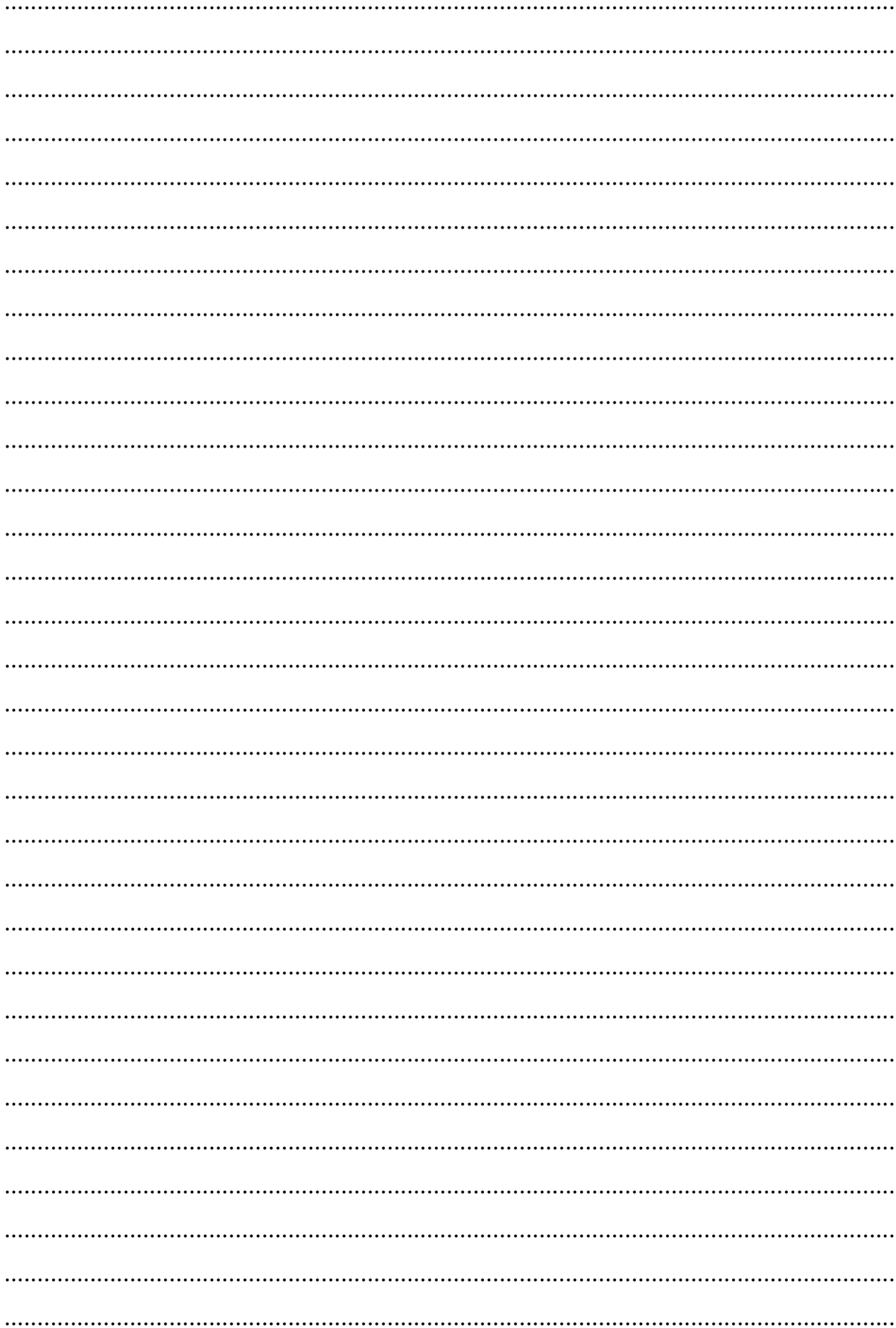
1. What is it?
2. What are the possible Social, Moral, Cultural and Ethical benefits of the technology on society
3. What are the possible Social, Moral, Cultural and Ethical risks of the technology on society
4. My conclusion on this technology and what it will mean for our world 10 years from now

Additional help:

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

SLR 17 – Ethical, morale and cultural issues

<https://student.craigndave.org/videos/slr-17-ethical-moral-and-cultural-issues>



What is “computational thinking”?

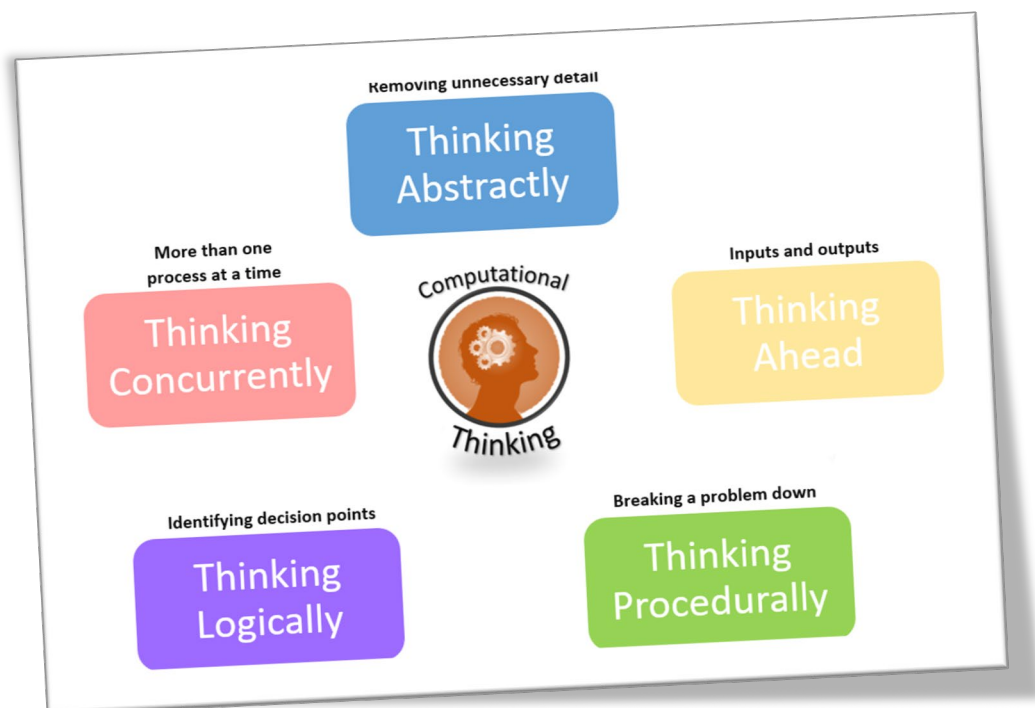
Thinking like a computer

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 involve a variety of “Computational Thinking” skills.

Using the Computational thinking and Computational method placemats. 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.

Your goal is to imagine someone else must revise from your mind map. Ask yourself:

- Does it make sense?
- Is it clear?
- Does it cover all the important concepts?



***Insert the computational thinking and computational methods
placemats***

Mind map for Computational thinking

Note taking practice task

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.

Using the Cornell note taking template on the following pages. Pick two of the following videos from Craig 'n' Dave:

- OCR: <https://student.craigdave.org/videos/ocr-alevel-slr01-alu-cu-registers-and-buses>
- OCR: <https://student.craigdave.org/videos/ocr-alevel-slr04-paging-segmentation-and-virtual-memory>
- OCR: <https://student.craigdave.org/videos/ocr-alevel-slr05-stages-of-compilation>
- OCR: <https://student.craigdave.org/videos/ocr-alevel-slr14-data-structures-part-2-graphs>

Write the title of the videos and its topic in the top boxes (use a different sheet for each video)

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 previous few minutes.

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 keywords and their definitions words the notes and list them in the bottom section.

insert cornell note making

Key terms task

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	Operating System
Von Neumann	Optical Storage	Compiler
Architecture	Device Driver	Lossy Compression
Intermediate Code	Machine Code	TCP/IP Stack
Assembly language	Normalisation	Problem
Hashing	ASCII	Decomposition
Packet Switching	Busses	

1. Research each of the key terms and write a definition
2. Resist the urge to simply copy 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?

Keyword	Definition
Control Unit	
Von Neumann Architecture	
Intermediate Code	
Assembly language	
Hashing	
Packet Switching	
Register	
Optical Storage	
Device Driver	
Machine Code	
Normalisation	

ASCII	
Busses	
Operating System	
Compiler	
Lossy Compression	
TCP/IP Stack	
Problem Decomposition	

An introduction to the basics of programming

Programming basics

Learning to “code” is a fun and essential part of the A-Level Computer Science.

1. Head over to the website: <https://www.learnpython.org/>
2. Complete the following python tutorials under the heading:
 - a. Hello, World!
 - b. Variables and Types
 - c. Lists
 - d. Basic Operators
 - e. String formatting
 - f. Basic string operations
 - g. Conditions
 - h. Loops
 - i. Functions
3. Each section presents you with theory, code to run and exercises to try out