Sixth Form Induction

SILVERDALE
SIXTH FORM

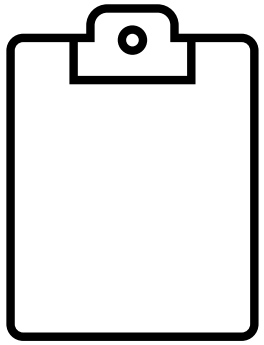
Computer Science

What content will be covered in the course?

- Internal computer components
- Cyber security
- Data representation
- Effect of digital technology on society
- Programming
- Networking and the Internet
- Software development

What skills will you need to be successful? How can you improve these skills?

- Resilience is the main skill required.
- Independent learning
- Discovery
- 100% effort



To improve all these skills you need to be working hard in and out of lessons. Look at current affairs and read around all subjects. Spend your time in lesson wisely. You should treat each lesson as an opportunity for clarification and issue fixing.

How will you be assessed during the course and in the final exams?

	Component 1: Computer systems	**Component 2: Computational thinking, algorithms and programming**	**Practical programming experience**
How is it assessed	Written exam 2 hour 30 minutes	Written exam 2 hour 30 minutes	Marked by your teacher and moderated by OCR.
How much is it worth	140 marks Worth 40%	140 marks Worth 40%	70 marks Worth 20%
Other information	A series of short-answer and extended-answer questions.	A series of short-answer and extended-answer questions.	Assesses student's ability to use the knowledge and skills gained through the course to solve or investigate a practical problem.

What will you need to organise your materials for the course?

- There is a list of essential and optional resources which will be available.
- You need a lever arch file folder to bring to every lesson.
- Make notes every lesson- Cornell Notes
- Organise your notes and lesson tasks.
- We have regular folder checks and you need to stay on top of it.

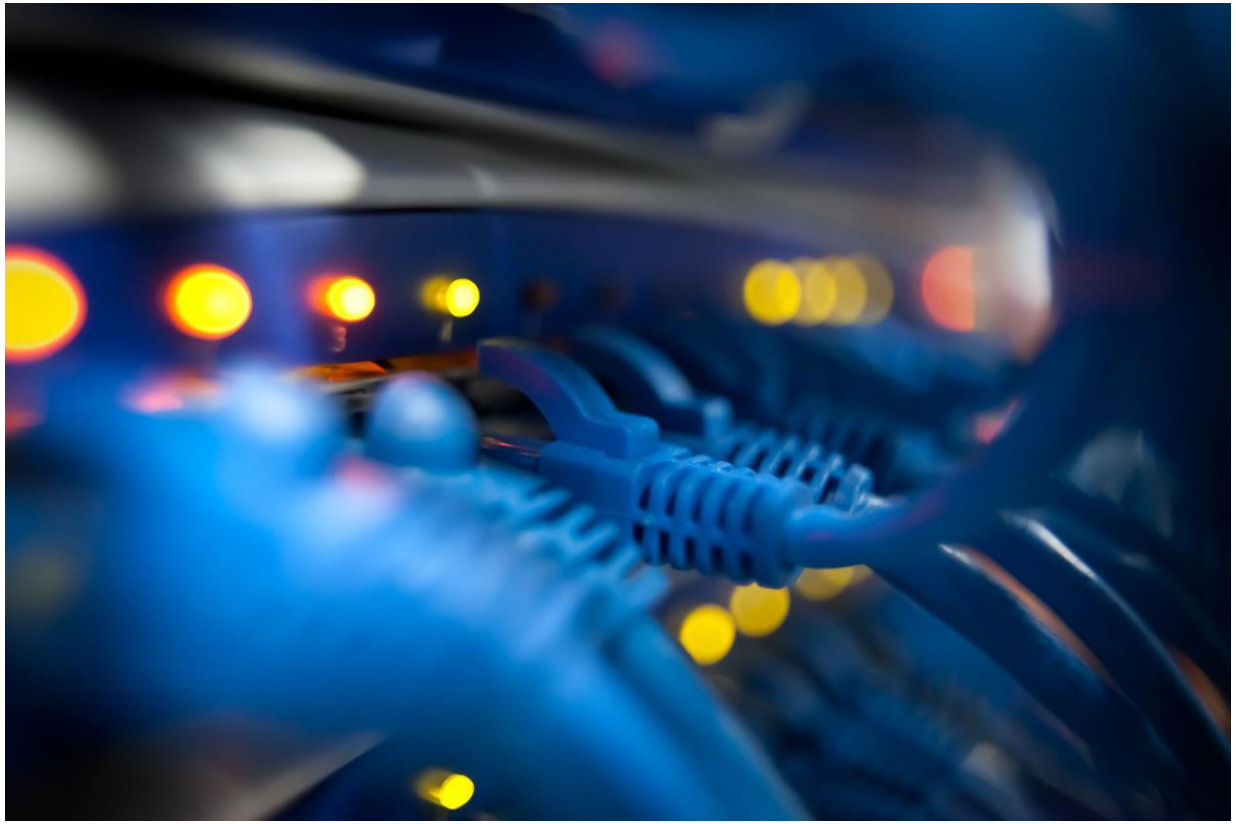
What is expected of you during the course?

- Effort
- You need to complete every lesson task, all note taking and every task provided to the best of your ability.
- You need to be asking questions in every lesson and not being a passive learner!

Bridging work

You must complete task 1, 11, 12 and 13.

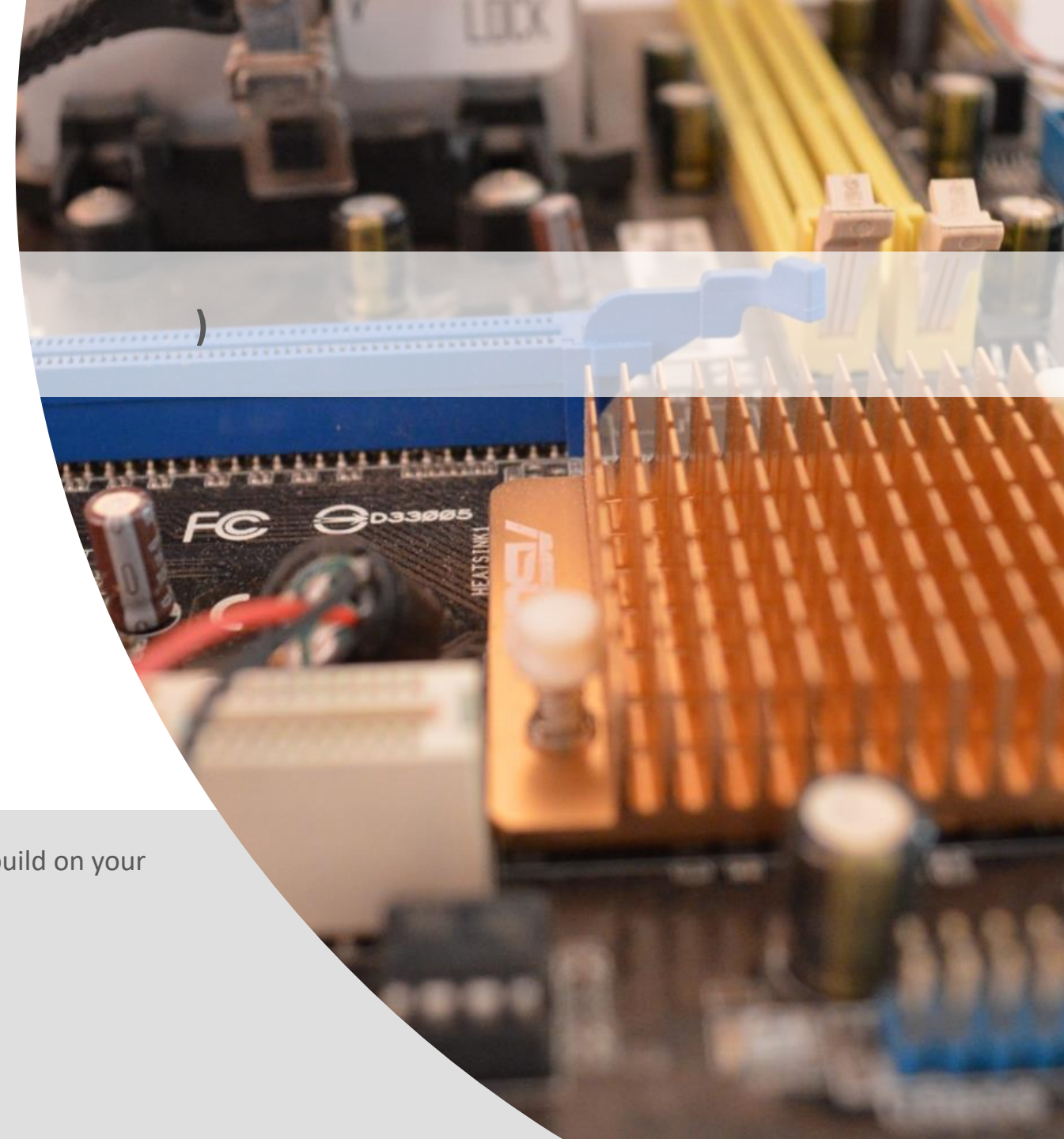
You may complete other tasks to a similar high standard,



GCSE to A level

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 todays 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.

Please complete by your first lesson back in September.

1 “Tell me about yourself”

Why did you choose Computer Science?

Expected time to complete: ½ hour

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 where you good at, what areas would you like to improve on?

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

2 Independent research task

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 cards
- 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://student.craigndave.org/videos/slr-17-ethical-moral-and-cultural-issues

3 What is “computational thinking”?

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.

1. Download the “Computational thinking and Computational methods placemats” from Craig n Dave:
   - https://student.craigndave.org/specification-key-terminology-and-cheat-sheets

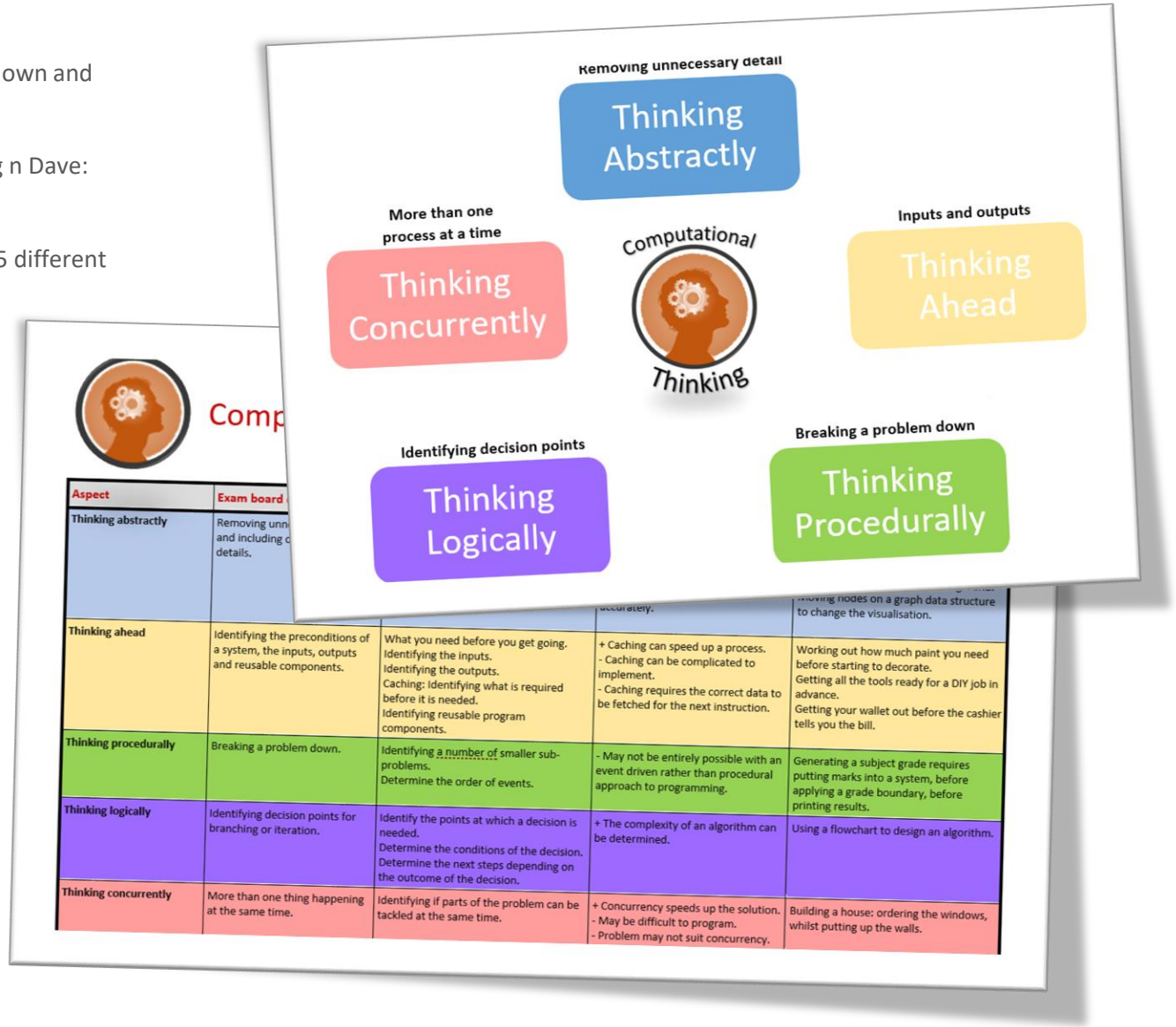
2. Create your own spider diagram / mind map which shows your clear understanding of the 5 different computational thinking strands
   - Keep it to a single side of A4 / A3

3. 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?

Note:

Although the five strands listed (and the download resources provided for this task) are from the OCR AS / A’Level specification, the concepts of “Computational Thinking” are just as applicable to the AQA course.

Indeed many of the strands listed are explicitly covered in the AQA specification in different locations.



Aspect	Exam board			
Thinking abstractly	Removing unn and including details.			to change the visualisation.
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.	+ Caching can speed up a process. - Caching can be complicated to implement. - Caching requires the correct data to be fetched for the next instruction.	Working out how much paint you need before starting to decorate. Getting all the tools ready for a DIY job in advance. Getting your wallet out before the cashier tells you the bill.
Thinking procedurally	Breaking a problem down.	Identifying a number of smaller sub-problems. Determine the order of events.	- May not be entirely possible with an event driven rather than procedural approach to programming.	Generating a subject grade requires putting marks into a system, before applying a grade boundary, before printing results.
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.	+ The complexity of an algorithm can be determined.	Using a flowchart to design an algorithm.
Thinking concurrently	More than one thing happening at the same time.	Identifying if parts of the problem can be tackled at the same time.	+ Concurrency speeds up the solution. - May be difficult to program. - Problem may not suit concurrency.	Building a house: ordering the windows, whilst putting up the walls.

4 Note taking practice task

The Cornell method of note taking

Expected time to complete: 1½ hours

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:
   - https://craigndave.org/product/cornell-note-taking-template/

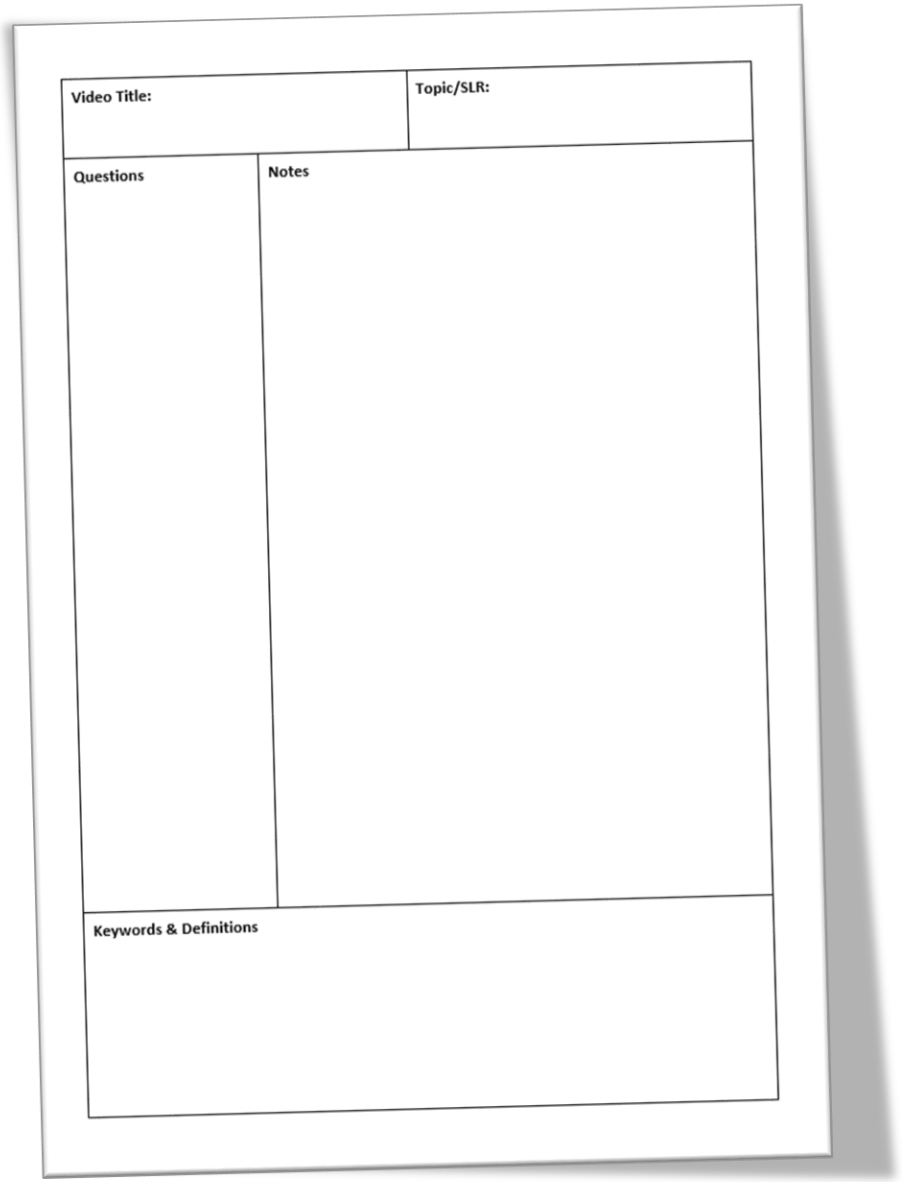
2. Pick any two of the following videos from Craig 'n' Dave:
   - OCR: https://student.craigndave.org/videos/ocr-alevel-slr01-alu-cu-registers-and-buses
   - OCR: https://student.craigndave.org/videos/ocr-alevel-slr04-paging-segmentation-and-virtual-memory
   - OCR: https://student.craigndave.org/videos/ocr-alevel-slr05-stages-of-compilation
   - OCR: https://student.craigndave.org/videos/ocr-alevel-slr14-data-structures-part-2-graphs

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.

6. Finally pull out all the key words and their definitions words the notes and list them in the bottom section.



Video Title:	Topic/SLR:
Questions	Notes
Keywords & Definitions	

5 Key terms task

Getting to grips with terminology

Expected time to complete: 2 hours

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?

6 An introduction to the basics of programming tasks

Programming basics

Expected time to complete: 6 hours

Learning to “code” is a fun and essential part of A Level Computer Science. This task is ideal if you haven't done the GCSE in Computer Science or you simply want a nice refresher ahead of starting your A Level course.

1. Head over to the web site: https://www.learnpython.org/
2. Complete the following python tutorials under the heading:
    - Hello, World!
    - Variables and Types
    - Lists
    - Basic Operators
    - String Formatting
    - Basic String Operations
    - Conditions
    - Loops
    - Functions
3. Each section presents you with theory, code to run and exercises to try out.
4. If you want to practice writing your own python programs you can download and install a simple python development tool here: https://www.python.org/downloads/

Additional note:

This task is most suited to students who intend to do the A Level and have not previously gained much / or any programming experience from the GCSE Computer Science course.

Although the language chosen here is Python, and that may not be what you will be using at A Level, it is the underlying programming concepts which are important.

The list of topics above cover the standard set of programming concepts you would be expected to know having completed a GCSE and Computer Science and so will prepare you well for the A level.

7 Systems software task

Operating systems

Expected time to complete: 2 hours

1. List at least 8 different roles an Operating System perform.

The purpose and roles of an Operating System

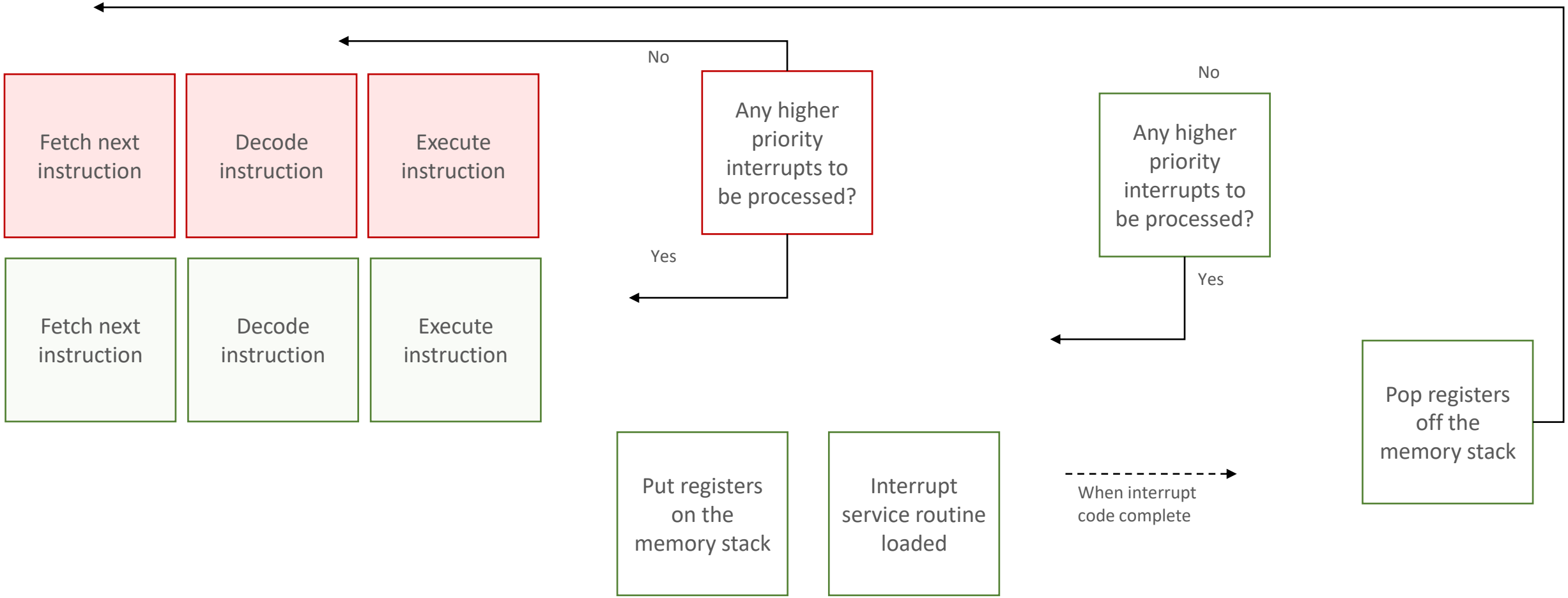
<Enter answer here>	<Enter answer here>	<Enter answer here>	<Enter answer here>	<Enter answer here>	<Enter answer here>	<Enter answer here>	<Enter answer here>

8 Systems software task

Operating systems

Expected time to complete: 2 hours

2. Rearrange the elements in the diagram below to show hoe the fetch-decode-execute cycle handles interrupts.



9 Computational logic task

Truth tables to circuit diagrams

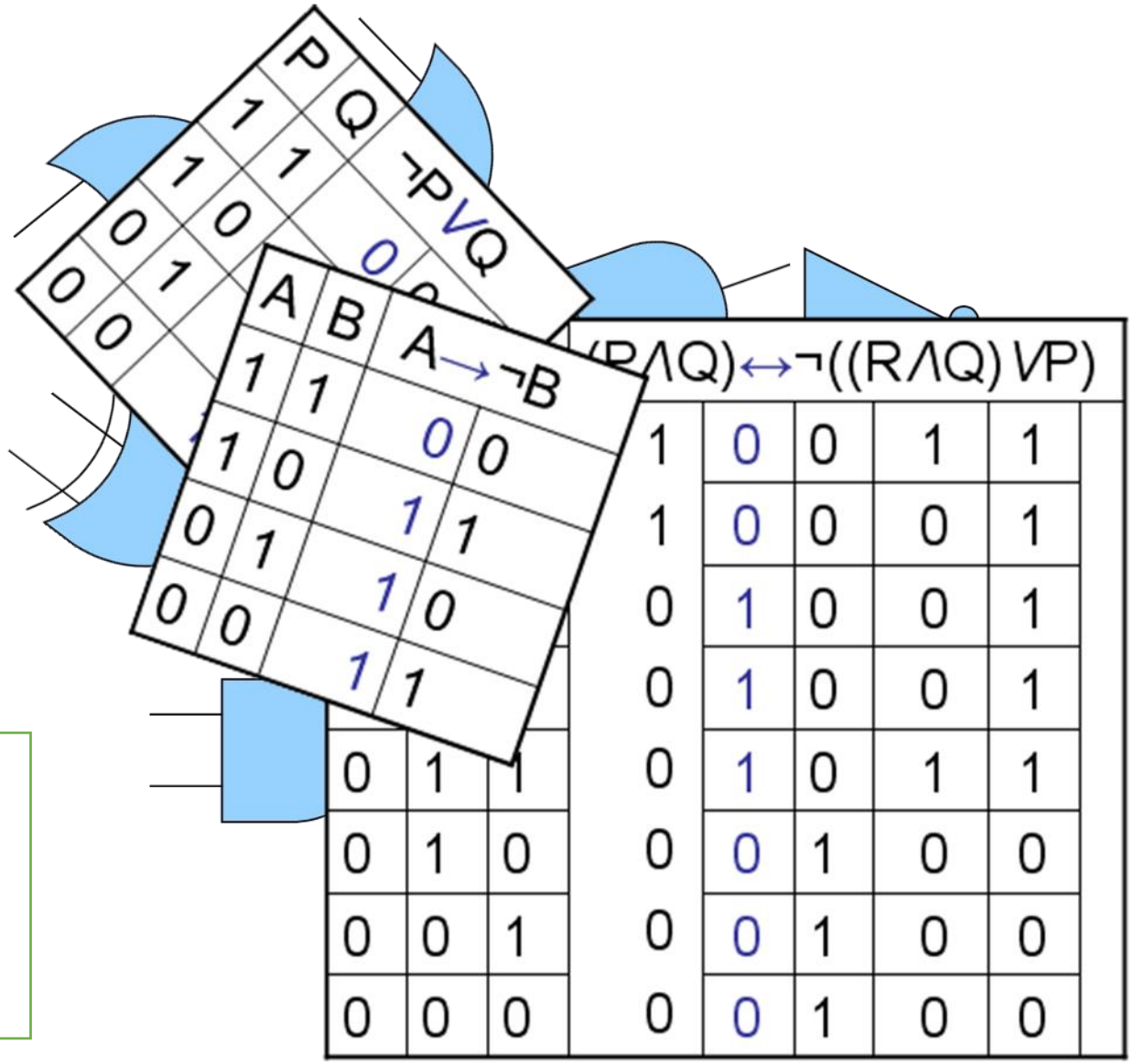
Expected time to complete: 2 hours

An important area of computer science is understanding the logic gates and diagrams which are used to represent the physical circuitry of computer systems.

Carry out some research into the following areas:

- Logic gates:
  - AND
  - NAND
  - NOR
  - NOT
  - OR
  - XOR
- Truth tables
- Boolean expressions
- Circuit diagrams

Complete the tasks on the following slides.



P	Q	¬P∨Q
1	1	0
1	0	
0	1	
0	0	

A	B	A→¬B
1	1	0
1	0	1
0	1	1
0	0	1

			(P∧Q)↔¬((R∧Q)∨P)				
			1	0	0	1	1
			1	0	0	0	1
			0	1	0	0	1
			0	1	0	0	1
0	1	1	0	1	0	1	1
0	1	0	0	0	1	0	0
0	0	1	0	0	1	0	0
0	0	0	0	0	1	0	0

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 15 – Boolean algebra
https://student.craigndave.org/videos/slr-15-boolean-algebra

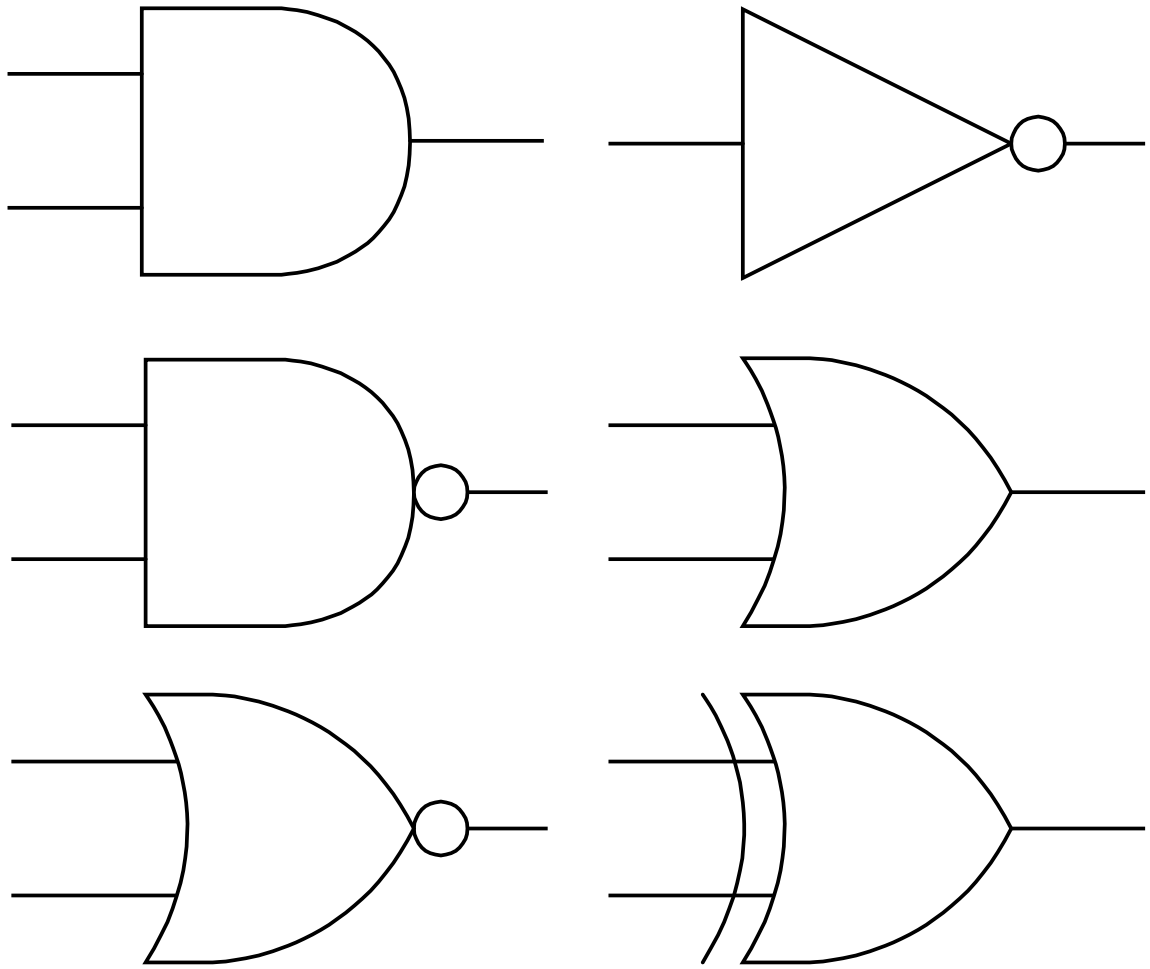
AQA: SLR16 – Logic gates & Boolean algebra
https://student.craigndave.org/videos/slr16-logic-gates-boolean-algebra

10 Computational logic task

Expected time to complete: 2 hours

Truth tables to circuit diagrams

1. Drag the labels into their correct place on the following diagram:



OR

AND

XOR

NOT

NAND

NOR

11 Computational logic task

Truth tables to circuit diagrams

Expected time to complete: 2 hours

2. Draw the circuit diagram which would represent the following Boolean expression:

OCR Boolean Expression: **F = ¬(¬C ∧ (A ∨ B)**

A

B

C

F

3. Complete the truth table for the circuit diagram you have drawn

A	B	C	D	E	F
0	0	0			
0	0	1			
0	1	0			
0	1	1			
1	0	0			
1	0	1			
1	1	0			
1	1	1			

12 Binary task

Expected time to complete: 1½ hours

Representing negative numbers in binary

In GCSE computer science you will have learnt how to represent positive whole numbers in binary e.g. 47

At A Level you will need to know how to represent negative as well e.g. -47

Start to recapping (or learning if you didn't do the GCSE) how to represent positive whole numbers between 0-255 in binary

Now research how to represent negative numbers in binary using the method known as:

- Two's complement

Complete the tasks on the following slides.

Additional help:

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

GCSE recap: How to represent positive binary values 0-255
https://student.craigndave.org/videos/aqa-gcse-slr13-number-bases

A Level: Representing negative binary values using Two's Complement
https://student.craigndave.org/videos/aqa-alevel-slr11-twos-complement

13 Binary task

Representing negative numbers in binary

Expected time to complete: 1½ hours

1. Write out the positive binary number 107, the answer should be displayed in 8 bits.

128	64	32	16	8	4	2	1	Binary weighting
								Binary value

2. Write out the negative binary number -107 using Two's Complement, the answer should be displayed in 8 bits.

								Binary weighting
								Binary value

3. How would you represent the lowest negative number possible using Two's Complement, given 8 bits.

								Binary weighting
								Binary value

4. How would you represent the largest positive number possible using Two's Complement, given 8 bits.

								Binary weighting
								Binary value

14 Binary task

Converting between base-2, base-10 and base-16

Expected time to complete: 1½ hours

As humans we have use the decimal or denary number system (base-10), made up of the unique digits 0-9.

Computer systems at the most basic level use only binary 1's and 0's (base-2).

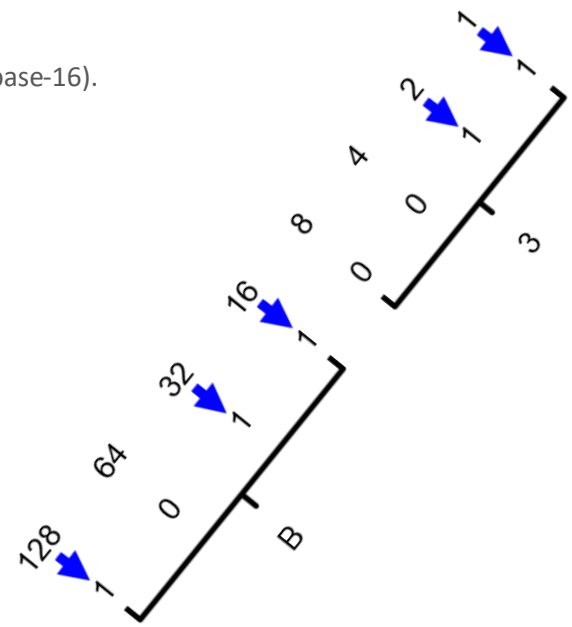
As a computer scientist you will also need to become familiar with the hexadecimal number system (base-16).

You will also need to be comfortable with converting numbers between these three base systems.

Research the following areas:

- Base-2 binary number system
- Base-10 decimal/denary number system
- Base-16 hexadecimal number system
- How to convert between base-2, base-10 and base-16

Complete the tasks on the following slides.



Denary	Binary	Hexadecimal
0	0000	0
1	0001	1
2	0010	2
3	0011	3
4	0100	4
5	0101	5
6	0110	6
7	0111	7
8	1000	8
9	1001	9
10	1010	A
11	1011	B
12	1100	C
13	1101	D
14	1110	E
15	1111	F

Additional help:

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

Base 2, 10 and 16 number systems:
https://student.craigndave.org/videos/aqa-alevel-slr10-base-2-10-and-16-number-systems

Converting between binary, hex and decimal:
https://student.craigndave.org/videos/aqa-alevel-slr11-aqa-converting-between-binary-hex-and-decimal

15 Binary task

Converting between base-2, base-10 and base-16

Expected time to complete: 1½ hours

1. Convert the base-2 binary number 11000101 into base-10 and base-16.
(Show all your working)

2. Convert the base-16 hexadecimal number 9F into base-2 and base-10.
(Show all your working)

You may wish to cut and paste this standard binary weighting line to help lay out parts of your answer

128	64	32	16	8	4	2	1	Binary weighting
								Binary value