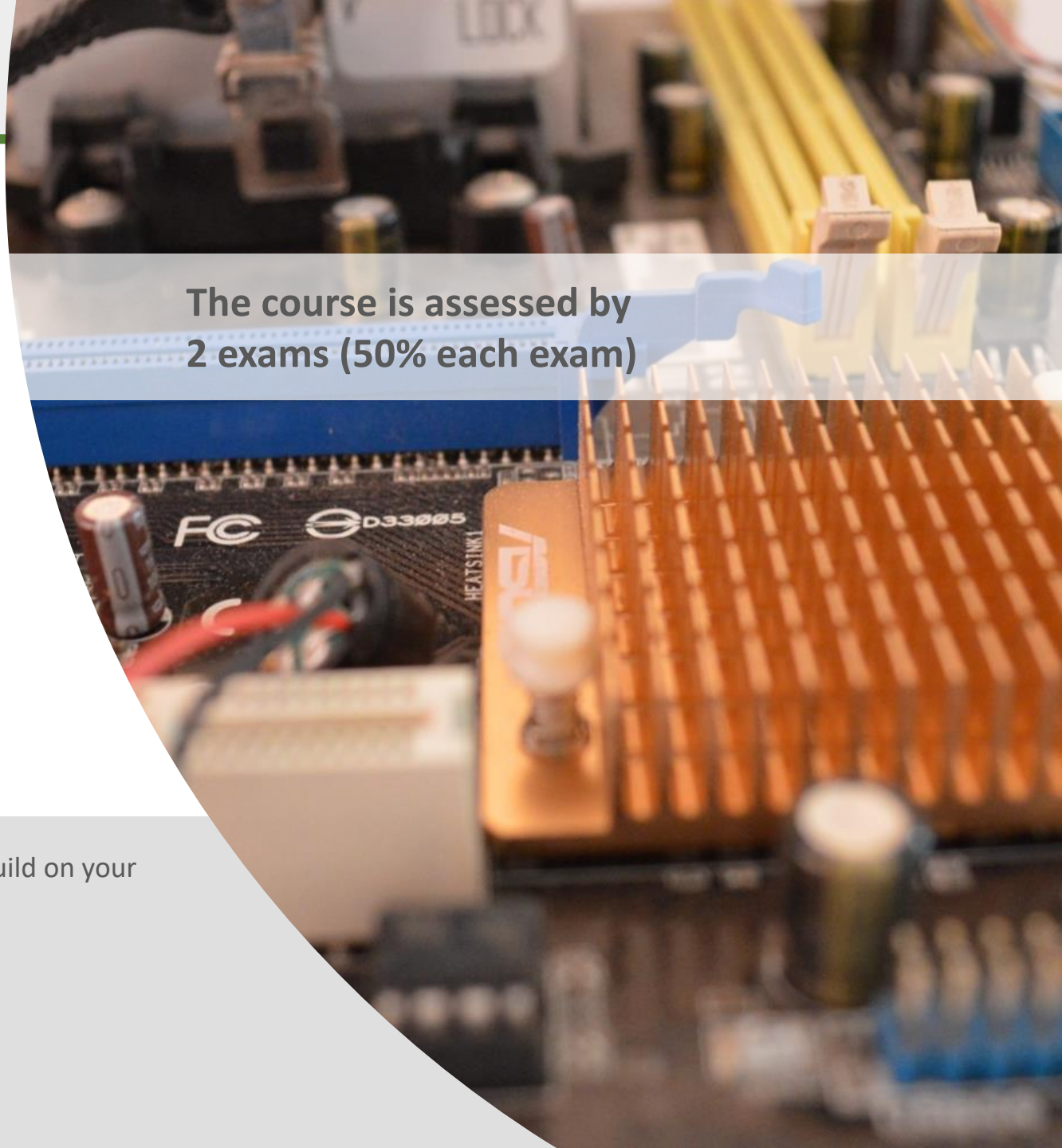# Computer Science Transition workbook

- 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**

The course is assessed by
2 exams (50% each exam)

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.

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?

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?
   - 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?

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.

Why is Computer Science important?

What is Computer Science?

- Enter your answer here

What are the benefits and risks of Computer Science at a local level

- Enter your answer here
- Try to make at least 4 valid points
- At least 2 of your points should be about the potential risks of Computer Science
- At least 2 of your points should be about the potential benefits of Computer Science

What are the benefits and risks of Computer Science at a national level

- Enter your answer here
- Try to make at least 4 valid points
- At least 2 of your points should be about the potential risks of Computer Science
- At least 2 of your points should be about the potential benefits of Computer Science

What are the benefits and risks of Computer Science at a global level

- Enter your answer here
- Try to make at least 4 valid points
- At least 2 of your points should be about the potential risks of Computer Science
- At least 2 of your points should be about the potential benefits of Computer Science

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.
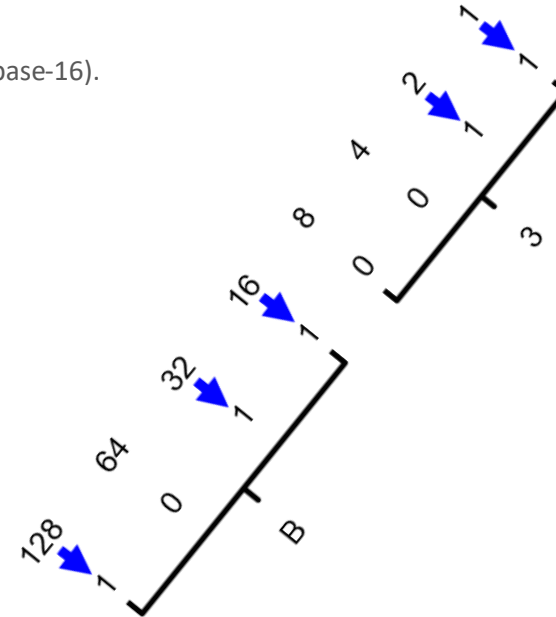
Additional help:

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

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

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

Algorithms: from theory to practice

Expected time to complete: 4 hours

A core concept of computer science is that of data structures and algorithms.

It is also an area which many students struggle with during examinations.

Probably the most basic algorithm is that of the "linear search".

If you have done the GCSE course you will have learnt about this searching algorithm already.

Start by learning or refreshing your knowledge of the linear search algorithm by using the videos on this page:
*   https://www.craigndave.org/algorithms-linear-search

Once you are happy with the theory complete the exercises on the following slides.

Algorithms: from theory to practice

Expected time to complete: 2 hours

1. Describe what the linear search algorithm does.

2. What are the applications of the linear search algorithm?

3. Write out the steps of the linear search algorithm in simple-structured English.

4. Draw a simple diagram which illustrates the linear search algorithm.

Algorithms: from theory to practice

Expected time to complete: 2 hours

5. Write out pseudocode for the linear search algorithm.

• The algorithm should use an array called items which is pre-populated with the following values: "Florida","Georgia","Delaware","Alabama","California"
• The algorithm should ask the user to "Enter the state to find:"
• If the algorithm locates the state entered by the user in the array it should report back to the screen "Item found at position n"
• If the algorithms can not locate the state entered by the user in the array it should report back to the screen "Item not found"

Algorithms: from theory to practice

Expected time to complete: 2 hours

6. Have a go at coding the linear searching algorithm in a programming language of your choice.

*   The program should work use an array called items which is pre-populated with the following values: "Florida","Georgia","Delaware","Alabama","California"
*   The program should ask the user to "Enter the state to find:"
*   If the program locates the state entered by the user in the array it should report back to the screen "Item found at position n"
*   If the program can not locate the state entered by the user in the array it should report back to the screen "Item not found"

Cut and paste the code you have written into the box below: