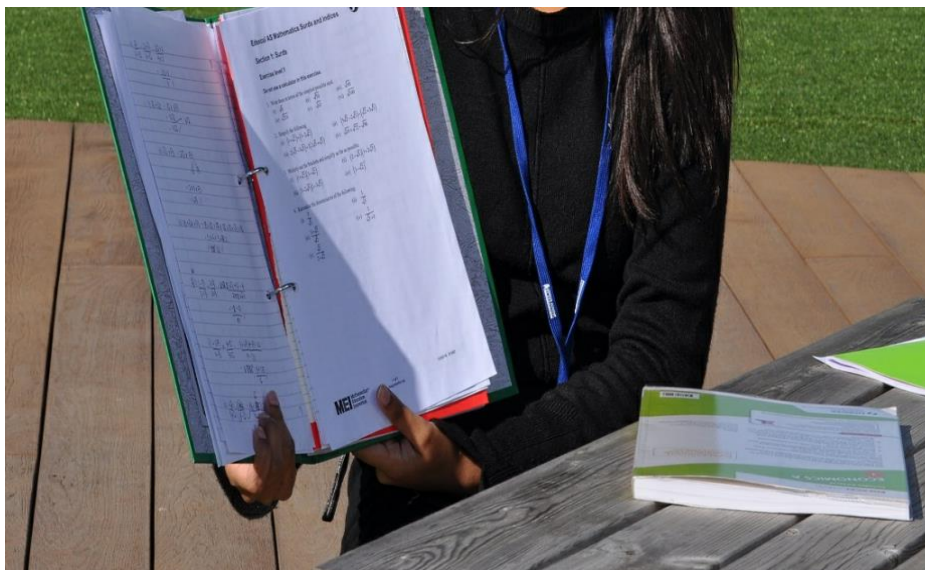




LONDON ACADEMY
OF EXCELLENCE

TOTTENHAM

The Place for Academic Rigour



LAE
Tottenham
Preparatory
Tasks for
Offer
Holders

COMPUTER SCIENCE

Preparatory Task: Computer Science



INTRODUCTION

A-Level Computer Science Content Overview

Component 01: Computer systems

Component 02: Algorithms and programming

Component 03: Programming project

TRANSITION ACTIVITIES

Computer Science Theory

1. Truth Tables – Task Sheet
2. Data types, data structures and algorithms
3. Memory
4. Storage
5. Networks

Algorithmic Thinking and Problem Solving

6. Puzzle 1

Writing code

7. Challenge 1.
8. Challenge 2.

RESOURCES

Recommended Reading

Recommended Websites

Algorithmic puzzles

INTRODUCTION

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.

The jump from GCSE to A level is huge and to be a successful Computer Scientist at A level it is essential you work on the key concepts such as Data representation, Data Structures and Searching and Sorting Algorithms. Have a go at the activities in this booklet. Answers will be provided for you to check your understanding. These activities also extend into KS5 content to give you a bit of a challenge. Go to the useful websites page to support you with these activities.

Once you have completed the transition activities, start to think about what programming project you would like to carry out and complete the analysis section.

You are expected to complete all work and bring it to your first lesson of Computer Science in September.

This transition pack is organised into four sections:

- Computer Science Theory.
- Algorithmic Thinking and Problem Solving.
- Writing Code.
- Useful Resources.

A-Level Computer Science Content Overview

Component 01: Computer systems

Students are introduced to the internal workings of the (CPU), data exchange, software development, data types and legal and ethical issues. The resulting knowledge and understanding will underpin their work in component 03.

It covers:

- The characteristics of contemporary processors, input, output and storage devices
- Types of software and the different methodologies used to develop software
- Data exchange between different systems
- Data types, data structures and algorithms

Component 02: Algorithms and programming

This builds on component 01 to include computational thinking and problem-solving.

It covers:

- What is meant by computational thinking (thinking abstractly, thinking ahead, thinking procedurally etc.)
- Problem-solving and programming – how computers and programs can be used to solve problems
- Algorithms and how they can be used to describe and solve problems.

Component 03: Programming project

Students are expected to apply the principles of computational thinking to a practical coding programming project. They will analyse, design, develop, test, evaluate and document a program written in a suitable programming language. The project is designed to be independently chosen by the student and provides them with the flexibility to investigate projects within the diverse field of computer science. We support a wide and diverse range of languages.

TRANSITION ACTIVITIES

Computer Science Theory

1. Truth Tables – Task Sheet

a) Write the truth tables for the expressions

NOT (A AND B)

A	B	A AND B	NOT (A AND B)

((NOT A) OR (NOT B))

A	B	NOT A	NOT B	((NOT A) OR (NOT B))

b) What do you notice about these tables

2. Data types, data structures and algorithms

No.	Denary	Binary	Hex	Binary value plus 00011110
1	1			
2	5			
3	10			
4	22			
5	40			

6	77			
---	----	--	--	--

b) Create a program that analyses a passage of text from a file and then counts:

- How many words
- The average length of a word
- How many times each word occurs
- How many words start with each letter of the alphabet?
- The aim of this exercise is to test your ability to develop algorithms.

3. Memory

- A. Compare RAM and ROM
- B. Explain the need for virtual memory in a computer system
- C. Describe the characteristics of flash memory

4. Storage

A. Complete the following table comparing optical, magnetic and solid state.

	<u>Capacity</u>	<u>Speed</u>	<u>Portability</u>	<u>Durability</u>	<u>Reliability</u>	<u>Cost</u>
Optical						
Magnetic						
Solid State						

B. Justify one use of each storage method

5. Networks

1. Explain the similarities and differences between:

- a. A LAN and a WAN

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

Algorithmic Thinking and Problem Solving

The following puzzle 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 puzzle is representative of classical problems and problem-solving strategies. You can solve each one by trial and error, but you encouraged to think about the strategy you employed to solve the problem.

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?

- 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 o 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.

6. Puzzle 1

The Princess in the Castle Originally heard on Puzzle Panel on BBC Radio 4.

<http://www.bbc.co.uk/programmes/b00xhd45>

A princess lives in a long corridor in a castle. The corridor has 17 rooms, numbered 1 to 17 inclusive. Each night the princess sleeps in a different room according to the following rules:

On the first night of the year, she sleeps in a random room

Each night she moves to an adjacent room; she never sleeps in the same room on two nights in a row and she always moves exactly one room left or right along the corridor

- For example, if she is currently sleeping in room 12, then on the next night she will either be in room 11 or in room 13
- If she is in room 1, then she must be in room 2 on the next night as she cannot move in any other direction (the same is true for room 17 – she must move to room 16 next)

A prince wishes to marry the princess. To do this he must find her room in the castle. However, whenever he sneaks into the castle at night, the guards quickly find him and throw him out! Therefore he only has time to search one room each night. The princess is unable to give the prince any clues to her location, and the prince has no knowledge of her location, other than whether or no she was in the room he last tried.

1. What strategy should the prince follow in order to find the princess in a finite time?
2. What is the maximum number of nights the prince needs to search before he can guarantee finding the princess?

Writing code

The coding challenges below will let you check your skills. Part of the transition to A-level is combining skills, and also ensuring that you plan and test your work thoroughly, so think about how you can re-use components and design your code for readability and robustness.

Write a program to:

- 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.

8. Challenge 2.

Apples and Bananas

Write a function, that replaces all vowels in a string with a specified vowel.

Examples

`vow_replace("apples and bananas", "u")` → "upplus und bununus"

`vow_replace("cheese casserole", "o")` → "chooso cossorolo"

`vow_replace("stuffed jalapeno poppers", "e")` → "steffed jelepene peppers"

Notes

All words will be lowercase. Y is not considered a vowel.

Recommended Reading

The following is a list of suggested texts that may help you with your studies of A-Level Computer Science, it is not an exhaustive list:

GCSE Computer Science textbooks

OCR AS and ALevel Computer Science books

- [AS/A Level Computer Science for OCR Student Book](#) from Cambridge University
- [OCR A Level Computer Science \(includes AS Level\)](#) from Hodder Education
- [OCR AS and A Level Computer Science](#) from PG Online Ltd

Recommended Websites

101 Computing	<p>Lots of facts and interactives to help you to understand some of the different topics covered in the A-Level course. There are also sections for the GCSE content as well if you want/need a refresher</p>
W3schools	<p>Tutorials and examples of code used for various different aspects of web development, such as HTML, CSS, JavaScript, SQL. Also has a “Try it” editor which allows you to write code and then see it in action within your web browser.</p>
<p>Codewars.com</p>	<p>Codewars is a website where you can practise your coding skills in many languages.</p>
Isaaccomputerscience	<p>Isaac Computer Science is a free online learning platform for A level, funded by the Department for Education.</p>
Craig n Dave	<p>Lots of videos and resources</p>

Algorithmic puzzles

Algorithmic Puzzles

A collection of puzzles designed to test and develop your algorithmic thinking and problem-solving strategies. The book is well organised, with a discussion of each problem-solving strategy and then several puzzles to practice.