Your notes

GCSE Computer Science: revision notes

Computational thinking and algorithms

Contents

1. Computational thinking
2. Algorithms
3. Structure diagrams and trace tables
4. Efficiency
5. Key terms
6. Quick questions
7. What to do next

Computational thinking

- **Decomposition** means breaking a problem down into smaller parts that are easier to solve.
- **Abstraction** means removing unnecessary detail so you can focus on what matters. A map of the London Underground shows the stations and how they connect, but not the real distances.
- **Algorithmic thinking** means working out the steps needed to solve a problem, in the right order.

Algorithms

- An **algorithm** is a **sequence** of steps that completes a task. A program is an **algorithm** written in a programming language, so a computer can run it.
- **Algorithms** are built from three things: **sequence** (steps in order), **selection** (choosing a path, such as with IF) and iteration (repeating steps with a loop).
- Before designing, identify the inputs, processes and outputs. For a program that works out the area of a rectangle, the inputs are the length and width, the process is length × width, and the output is the area.

Structure diagrams and trace tables

- A **structure diagram** shows a problem broken down into smaller sub-problems, from the top down.
- A **trace table** records the value of each variable as an **algorithm** runs, step by step. Working through an **algorithm** by hand like this (a dry run) helps you find errors.

Efficiency

- More than one **algorithm** can solve the same problem. A more efficient **algorithm** takes fewer steps (less time) or uses less memory.
- For example, a binary search finds an item in a large sorted list in far fewer steps than a linear search.

Key terms

Algorithm	A sequence of steps that completes a task.
Decomposition	Breaking a problem down into smaller, more manageable parts.
Abstraction	Removing unnecessary detail to focus on the important parts of a problem.
Algorithmic thinking	Working out the steps needed to solve a problem.
Sequence	Instructions carried out one after another, in order.
Selection	Choosing which path to follow, depending on a condition.
Iteration	Repeating a set of instructions.
Trace table	A table that records the values of variables as an algorithm runs.
Structure diagram	A diagram showing a problem broken down into sub-problems.

Quick questions

Cover the answers and test yourself: remembering is what makes it stick.

1. What is decomposition?
2. What is abstraction?
3. A Tube map shows stations and lines but not real distances. Which technique is this?
4. What is an algorithm?
5. Which three building blocks are all algorithms made from?
6. What is selection?
7. What is iteration?
8. What is a trace table used for?

Answers

1. Breaking a problem into smaller, more manageable parts
2. Removing unnecessary detail to focus on what matters
3. Abstraction
4. A sequence of steps that completes a task
5. Sequence, selection and iteration
6. Choosing which path to follow, for example with IF

Your notes

7. Repeating a set of steps
8. Recording the values of variables as an algorithm runs

What to do next

Practise 12 questions on this topic

Try 5 exam questions, with mark schemes

Read the lesson, one step at a time

Free revision for GCSE and KS3 at lycolearn.co.uk. Some notes are for one exam board only, and say so in brackets.

Your notes