

Computational Thinking

Key Revision Points

Objectives

Understand computational thinking including

- abstraction
- decomposition
- algorithmic thinking

Understand computational thinking including

- Make sure you revise the answer **S** to the question
i.e. one answer in 3 parts

Computational thinking

- Computer Science is about studying problems and how to solve them
 - How can you route pieces of information across a network to the other side of the world?
 - How can you make the images in a computer game look more realistic?
 - How can you program a computer to work out the school timetable?
 - How can you search 1,000,000 items quickly?

What is an algorithm?



A set of instructions

- An algorithm is a set of instructions for solving a problem or completing a task
- The task could be:
 - Making a chocolate cake
 - Summing the numbers 1 to 1000
 - Building a Lego model
 - Think of some more ...



Algorithmic thinking

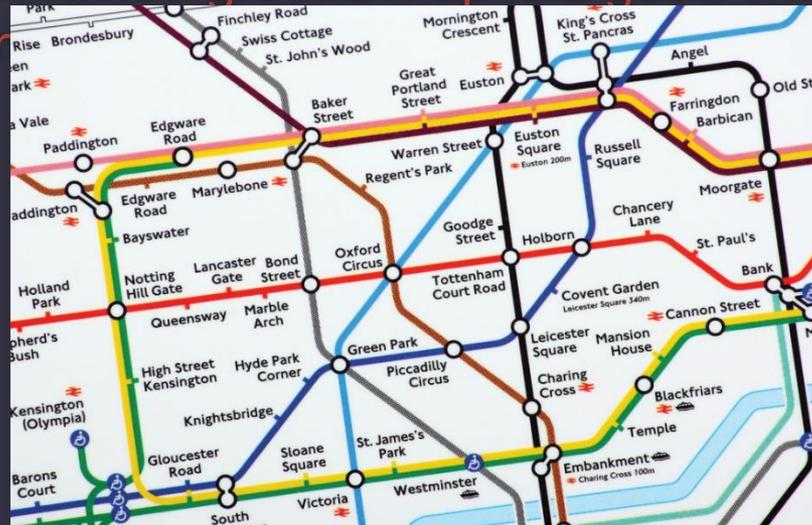
- Solving these puzzles involves **algorithmic thinking**
- If you are using a computer to find the answer, you have to figure out how to solve the problem, and then write down the steps
 - **Not all solutions are equally efficient!**

Divide and conquer

- Here is a problem you can solve by thinking algorithmically
- Ask a friend to think of a number between 1 and 1000
- Guess the number by asking: “Is the number greater than n (where n is your guess)”
 - How many guesses will you need to find the number?

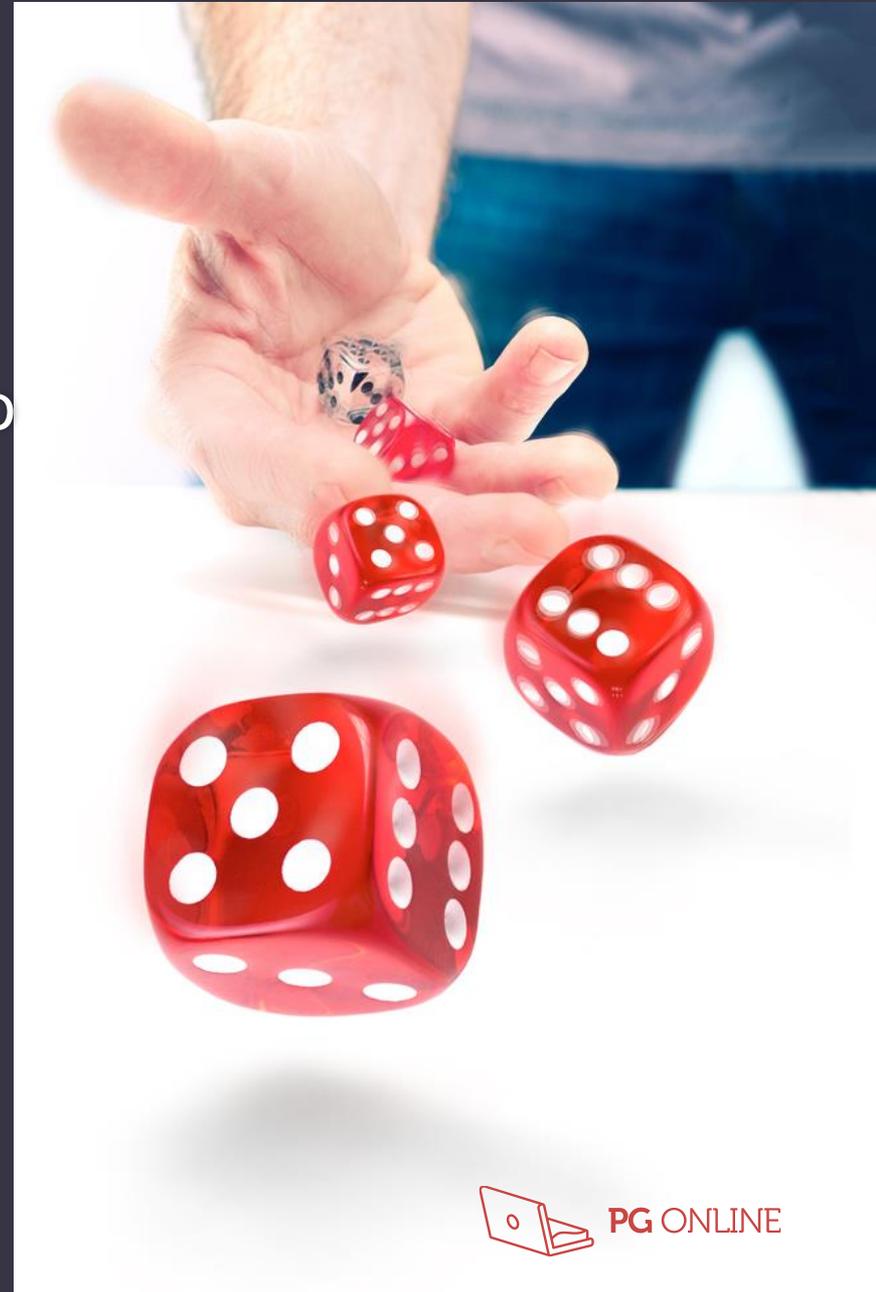
Abstraction

- Abstraction involves removing unnecessary detail from a problem so that you can focus on the essentials
- The London Underground map is a good example of abstraction



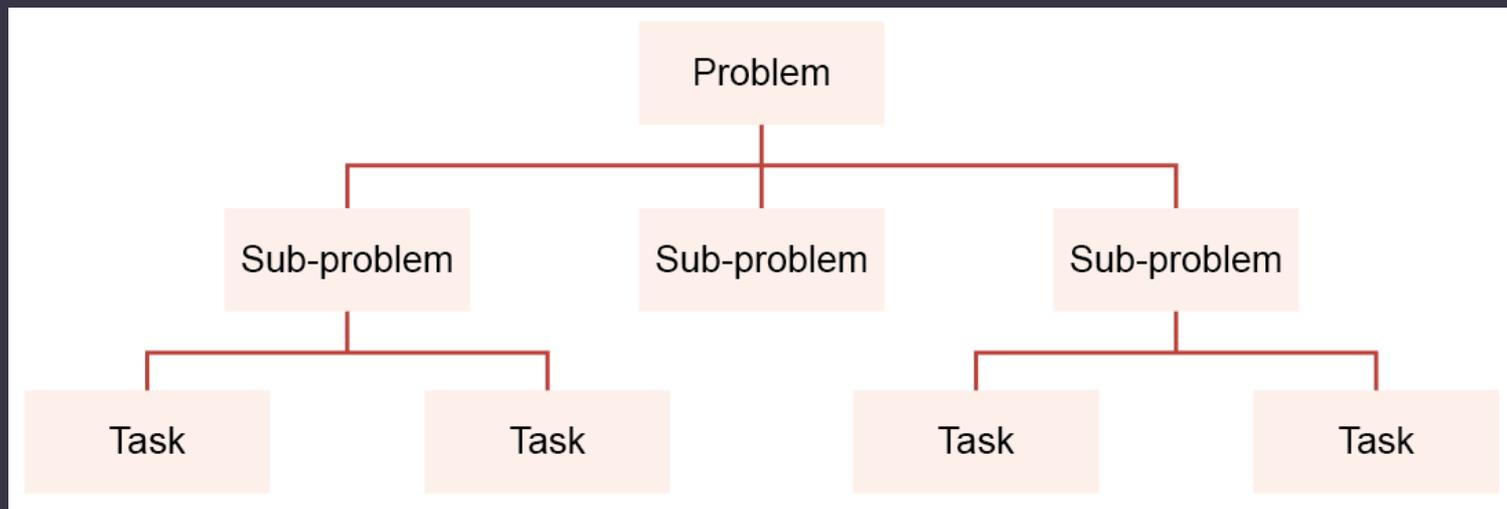
Abstraction

- When you write a program to play a game involving dice with a computer, how does the computer “roll the dice”?



Decomposition

- Decomposition involves breaking down a large problem into smaller sub-problems
- Then the sub-problems can be broken down further until each small task is manageable

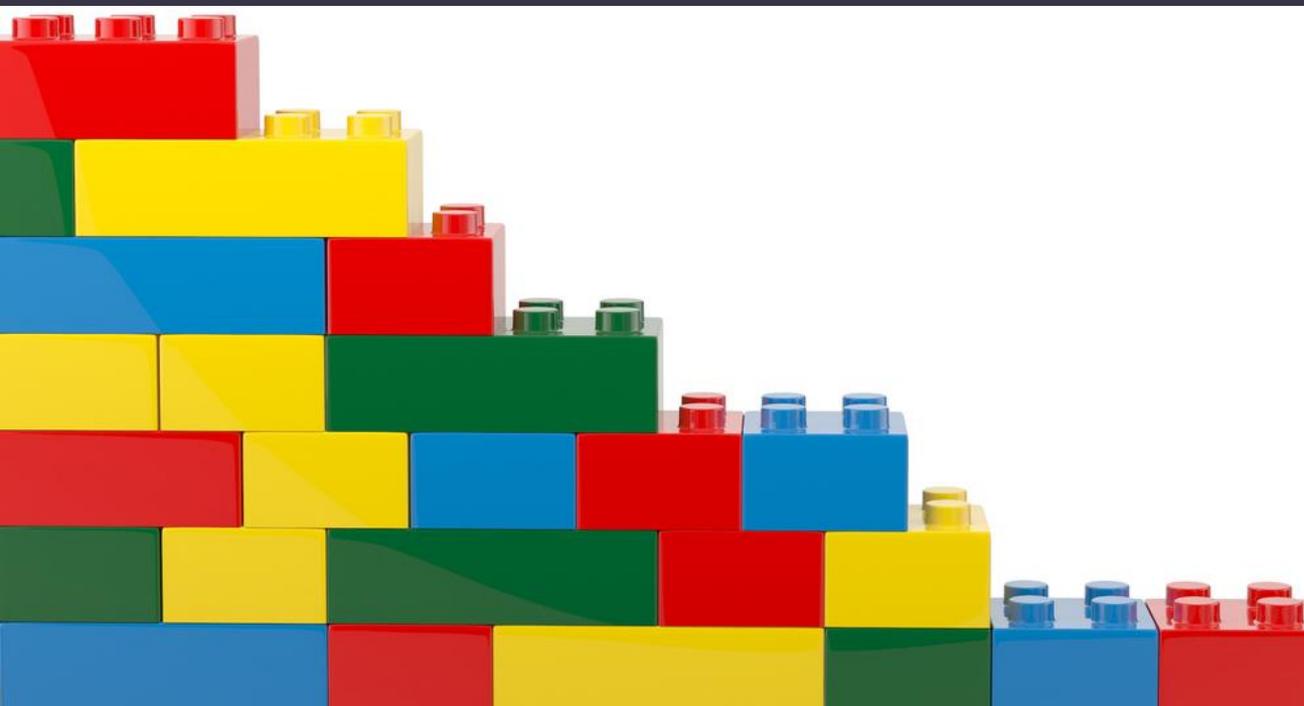


Decomposing a dice game

- Suppose you want to create a dice game to be played on the computer
- You need to think of the main tasks that need to be performed – for example:
 - Display the rules
 - Establish whether this particular game is to be a two-player game, or one person against the computer
 - Display the “board” if there is one
 - Play the game
 - Display the result when the game is over

Decomposition - advantages

- The problem becomes easier to solve when it consists of a number of small subtasks or **modules**



other programs,



Worksheet 1

- Try **Task 2** on the worksheet



Questions

1. Define the term **algorithm**
2. Explain briefly what is meant by **decomposition**
3. Explain the term **abstraction**
4. Explain how **abstraction** and **decomposition** are used in the design of an algorithm to calculate how much to charge for lemonade at a lemonade stand

Answers - 1

- An **algorithm** is a sequence of steps that can be followed to complete a task
- Note that:
 - An algorithm is **not** a computer program
 - A computer program is an **implementation** of an algorithm

Answers - 2

- Decomposition means breaking a problem down into a number of sub-problems
- Each sub-problem accomplishes an identifiable task, which might itself be further subdivided

Answers – Question 4

- Abstraction is used to remove details such as what product is being sold, what it tastes like, where the stand is
- The relevant features are identified – cost of production, amount of lemonade per cup etc.
- Decomposition is used to break the problem down:
 - Calculate cost of product
 - Estimate likely sales
 - Calculate sales for different scenarios