

# A LEVEL COMPUTER SCIENCE

## Lesson Element

## Thinking Abstractly

## *Instructions and answers for teachers*

*These instructions should accompany the OCR resource 'Thinking Abstractly' activity which supports OCR A Level Computer Science.*

**A LEVEL  
COMPUTER SCIENCE**

Lesson Element

Thinking Abstractly

Activity 1

Colour the map above, making sure that countries that share a border are not the same colour. You should also try and use as few colours as possible.

How many colours did you need?

How can you be sure?

**Extension – adding rules ('constraints')**

Countries 4 and 5 have now become one country. Does this change the graph and the colouring?

Country 3 does not exist anymore, and a new constraint has been introduced, that you must only have two colours. Is this possible?

Version 1

CAMBRIDGE UNIVERSITY PRESS  
www.cambridge.org

OCR  
Oxford Cambridge and RSA

### **The Activity:**

These activities support an introduction to the computer science notion of abstraction.



*This activity offers an opportunity for English skills development.*

### **Associated materials:**

'Thinking Abstractly' Lesson Element learner activity sheet.



# A LEVEL COMPUTER SCIENCE

## Introduction

Abstraction is the ability to filter the details that we do not need out of a problem so that we can create a more general idea of what the problem is. We may then be able to go on and apply the same problem solving procedure to other problems.

For instance, you could give a list of each London underground station and which other stations they are connected to, or you could produce a map like the London Underground tube map that abstracts the problem, making it easier to understand. Since this was so famous, this is now a common 'template' to use for other underground mapping systems.

The first activity seems like a simple colouring activity. The activity will help students consider how a computer might solve this problem. They will then use the same abstraction to solve activity 2.

### Learning Outcomes:

After the activity, pupils should:

- Have experience in abstracting problems
- Recognise the advantages of why we abstract problems to 're-use' a solution
- Recognise that constraints make problems harder to solve
- Understand what graphs are and why they are used in computer science
- Understand why there is a need to automate tasks



# A LEVEL COMPUTER SCIENCE

## Activity 1

Introduce the idea of abstraction in computer science

- Pupils are asked to look at Activity 1 and colour it in using as few colours as possible.
- You can ask pupils how they achieved the task and see if there are any common themes.
- Ask questions focused on extending the problem and introducing graphs as a method of problem representation.
- Discuss the idea of constraints. Did anyone attempt the extension activities?

## Solution

First go through each node and list what other nodes it is connected to

1 – 2,3,6

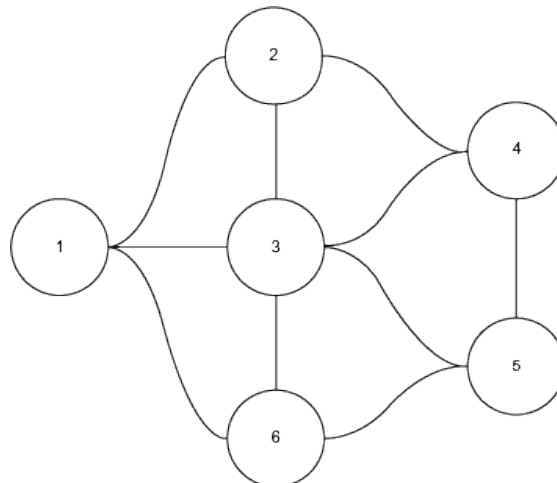
2 – 1,3,4

3 – 1,2,4,5,6

4 – 2,3,5

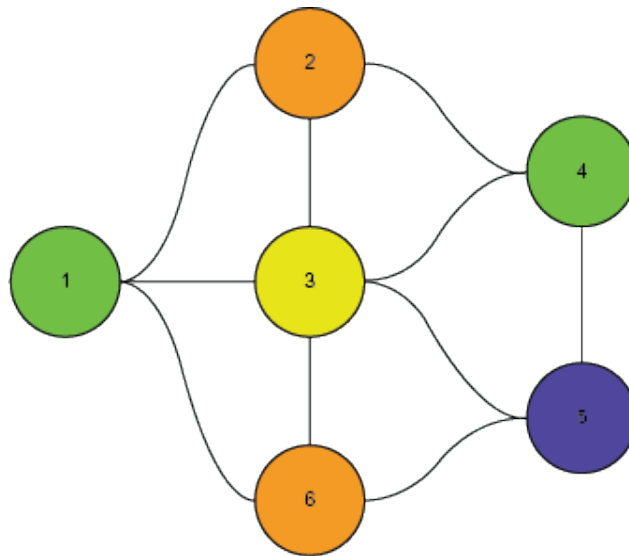
5 – 3,4,6

6 – 1,3,5



# A LEVEL COMPUTER SCIENCE

Start with one node and keep applying the rules



# A LEVEL COMPUTER SCIENCE

## Activity 2

Pupils should have a look at Activity 2. Can they figure it out for themselves?

### Extension (10 mins)

Students who finish early should be encouraged to explore their own graphs

### Discussion and conclusions (10 mins)

- Discuss with students what the idea of abstraction means to them and sort out any misconceptions.
- Discuss the use of rules to complete the tasks.
- Discuss limitations of using graphs, and that not all problems can be solved in this way.
- Discuss that as you may have found from Activity 2, computer logic does not cover all bases.

### Supporting links/Other activities

<http://csi.dcs.gla.ac.uk/workshop-view.php?workshopID=6> – Graph colouring. Materials for this exercise were adapted from this exercise for easier use in a lesson. Author: CS inside

<http://studio.code.org/s/1/level/81> - Abstraction madlibs activity. Author: code.org

<http://codeboom.wordpress.com/2012/11/22/python-madlibs/> - Python madlibs Author: Laura @codeboom

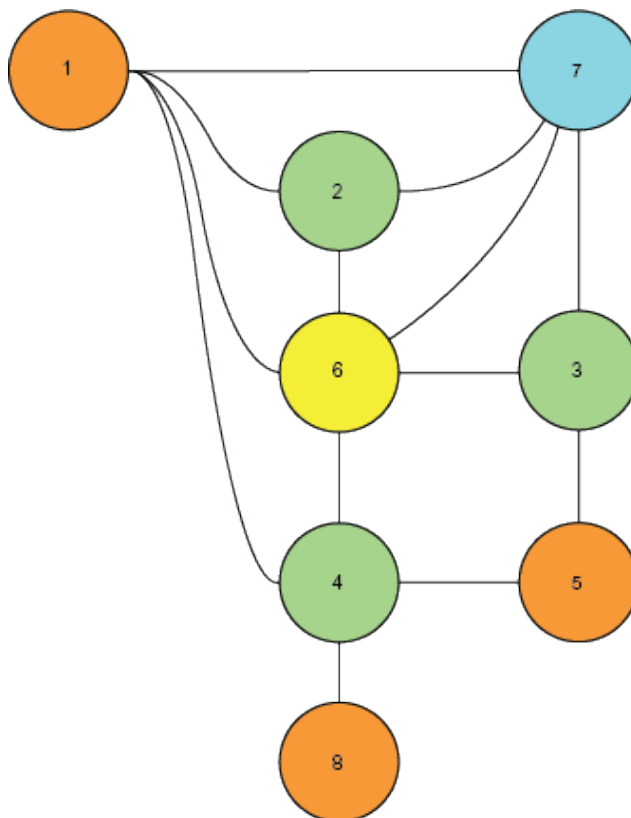
<http://www.bbc.co.uk/education/guides/zttrcdm/revision> - Abstraction : BBC Bitesize. Good overview, has a quiz.



# A LEVEL COMPUTER SCIENCE

## Solution

Of course, this problem can also be represented as a graph. It can be easier to draw if students replace the names of the animals with the numbers, and then they can work at each one in a logical order.



Of course, the problem is with the chart we have made is that it suggests that just because animals don't eat each other, that doesn't mean that they are suited to be living together. Computer science needs to have human logic attached to it too!



We'd like to know your view on the resources we produce. By clicking on the 'Like' or 'Dislike' button you can help us to ensure that our resources work for you. When the email template pops up please add additional comments if you wish and then just click 'Send'. Thank you.

If you do not currently offer this OCR qualification but would like to do so, please complete the Expression of Interest Form which can be found here: [www.ocr.org.uk/expression-of-interest](http://www.ocr.org.uk/expression-of-interest)

### OCR Resources: the small print

OCR's resources are provided to support the teaching of OCR specifications, but in no way constitute an endorsed teaching method that is required by the Board, and the decision to use them lies with the individual teacher. Whilst every effort is made to ensure the accuracy of the content, OCR cannot be held responsible for any errors or omissions within these resources. We update our resources on a regular basis, so please check the OCR website to ensure you have the most up to date version.

© OCR 2015 - This resource may be freely copied and distributed, as long as the OCR logo and this message remain intact and OCR is acknowledged as the originator of this work.

Please get in touch if you want to discuss the accessibility of resources we offer to support delivery of our qualifications [resources.feedback@ocr.org.uk](mailto:resources.feedback@ocr.org.uk)