

Accredited

A LEVEL

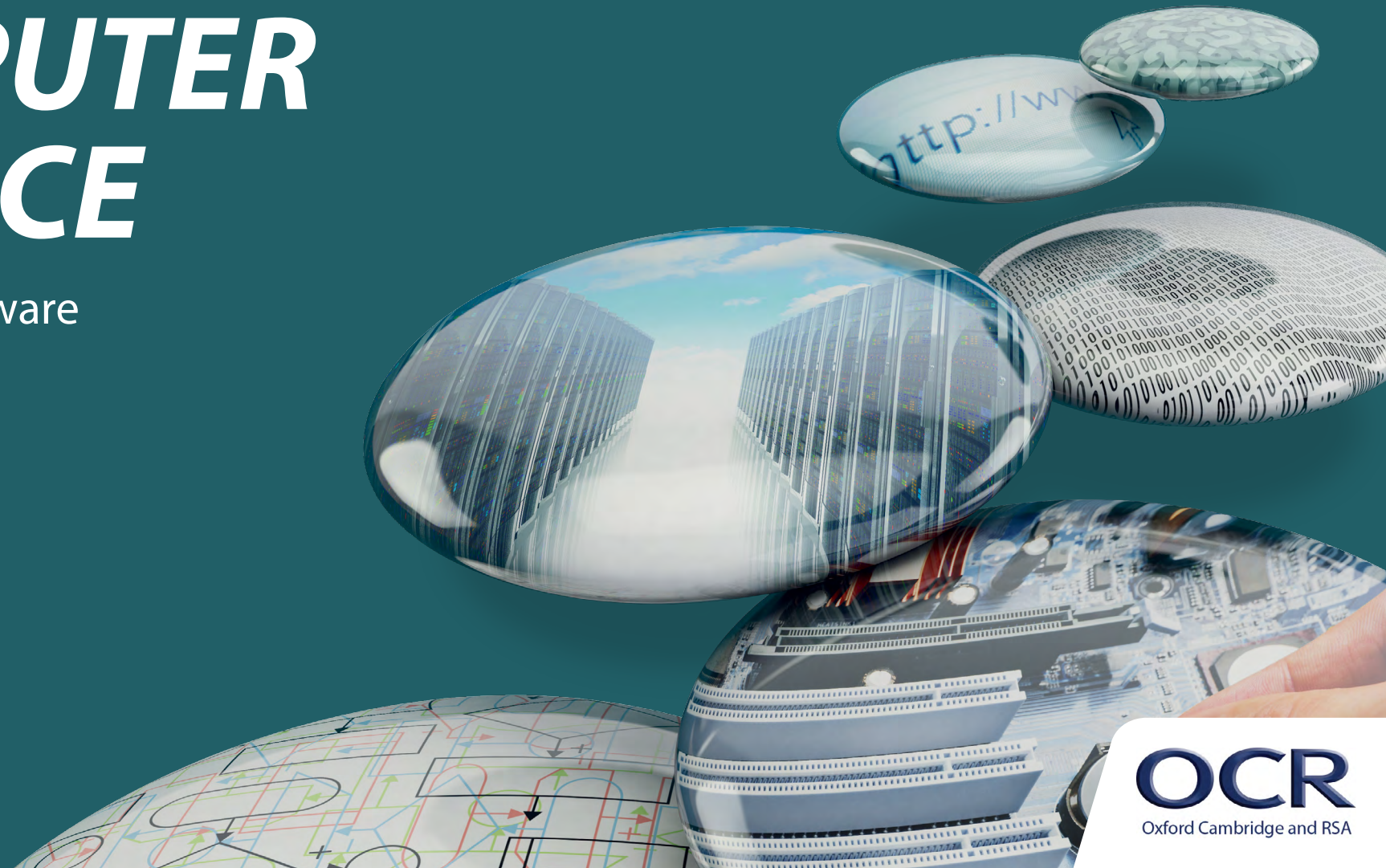
Delivery Guide

H446

COMPUTER SCIENCE

Theme: 1.2.3 Software
Development

June 2015



OCR
Oxford Cambridge and RSA

We will inform centres about any changes to the specification. We will also publish changes on our website. The latest version of our specification will always be the one on our website (www.ocr.org.uk) and this may differ from printed versions.

Copyright © 2015 OCR. All rights reserved.

Copyright

OCR retains the copyright on all its publications, including the specifications. However, registered centres for OCR are permitted to copy material from this specification booklet for their own internal use.

Oxford Cambridge and RSA Examinations is a Company Limited by Guarantee. Registered in England. Registered company number 3484466.

Registered office: 1 Hills Road
Cambridge
CB1 2EU

OCR is an exempt charity.

CONTENTS

Introduction	Page 4
Curriculum Content	Page 5
Thinking Conceptually	Page 6
Thinking Contextually	Page 8



Introduction

Delivery guides are designed to represent a body of knowledge about teaching a particular topic and contain:

- Content: a clear outline of the content covered by the delivery guide;
- Thinking Conceptually: expert guidance on the key concepts involved, common difficulties students may have, approaches to teaching that can help students understand these concepts and how this topic links conceptually to other areas of the subject;
- Thinking Contextually: a range of suggested teaching activities using a variety of themes so that different activities can be selected that best suit particular classes, learning styles or teaching approaches.

If you have any feedback on this Delivery Guide or suggestions for other resources you would like OCR to develop, please email resources.feedback@ocr.org.uk.

KEY



Click to view associated resources within this document.



Click to view external resources



AS Level content only



Curriculum Content

- a) Understand the waterfall lifecycle, agile methodologies, extreme programming, the spiral model and rapid application development.
- b) The relative merits and drawbacks of different methodologies and when they might be used.
- c) Writing and following algorithms.



Thinking Conceptually

Approaches to teaching the content


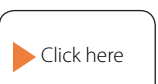
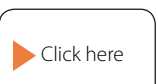
Students might not have come across the different software development methodologies as they usually work alone. A few activities are detailed which give practical examples of how some of the processes work.

As suggested in one of the activities, you could invite in a speaker from the industry to talk about how they work and deal with their software development processes; this would be a valuable insight for students to have.

Understand the waterfall lifecycle, agile methodologies, extreme programming, the spiral model and rapid application development	Resources
<p>The relative merits and drawbacks of different methodologies and when they might be used</p> <p>Waterfall model (systems life cycle) – This process is sequential and goes through the phases of conception, initiation, analysis, design, construction, testing, implementation and maintenance. It was first invented for hardware manufacture, but then was adapted for software development. http://www.teach-ict.com/as_a2_ict_new/ocr/A2_G063/331_systems_cycle/slc_stages/miniweb/index.htm</p> <p>Agile methodologies (agile software development) are a way of developing software in incremental parts called ‘sprints’ that typically last around three weeks. Therefore a website may be launched and then every three weeks new features are added. This helps to continually refine the requirements specification to the customer’s wants and needs.</p> <p>A good video that explains agile development can be found at: https://www.youtube.com/watch?v=OJfIDE6OaSc</p> <p>Extreme programming is a type of agile software development process, but it puts emphasis on pair programming, code review (part of pair programming) and unit testing. http://en.wikipedia.org/wiki/Extreme_programming</p> <p>The spiral model combines the idea of iterative development with the systematic aspects of the waterfall model. The spiral model has four phases (identification, design, build and evaluation/risk analysis). Each time the cycle is done the customer supplies feedback and risk analysis looks at the possibility of overruns of time/money requirements. http://www.tutorialspoint.com/sdlc/sdlc_spiral_model.htm</p> <p>Rapid application development is a software development process that relies heavily on programming and does not go into too much detail/time on requirements or planning. This needs heavy involvement from the customer to continually review prototypes to suggest improvements. http://www.teach-ict.com/as_a2_ict_new/ocr/A2_G063/331_systems_cycle/prototyping_RAD/miniweb/pg7.htm</p>	<p>▶ Click here</p> <p>▶ Click here</p> <p>▶ Click here</p> <p>▶ Click here</p> <p>▶ Click here</p>



Thinking Conceptually

Writing and following algorithms	Resources
<p>Algorithms are some of the most fundamental parts of most programs, especially where there is a lot of data involved. Algorithms are a set of rules to be followed in a problem-solving operation, so it is important that our 'rules' are appropriate for what we are trying to achieve.</p>	<p></p>
<p>The website below has a very good introductory video complete with activities and some in-depth information on types of sorting algorithm. http://www.csfieldguide.org.nz/Algorithms.html</p>	<p></p>
<p>Some good activities for practically illustrating different algorithms are given at the end of the teacher's guide.</p>	
<p>The website below covers Linear search, Binary search, Merging, Sorting, Insertion sort, Quicksort. http://www.teach-ict.com/as_as_computing/ocr/H447/F453/3_3_5/data_structures/miniweb_search/index.htm</p>	<p></p>
<p>There is also an interesting set of animations showing how fast different sorting algorithms are in comparison to each other: http://www.sorting-algorithms.com/</p>	

Common misconceptions or difficulties students may have




Students may have difficulties remembering which software development strategy is which. Diagrams will usually aid with this. The website <http://www.tutorialspoint.com/sdlc/index.htm> helps as it gives a clear definition of each process with an accompanying diagram.

Conceptual links to other areas of the specification – useful ways to approach this topic to set students up for topics later in the course

Students will gain experience with going through the stages of a project for their controlled assessment, but won't have exposure to working in a team environment for this.





Thinking Contextually

Activities	Resources
<p>Activity 1 – Fun with the waterfall process (CSFieldguide) http://www.csfieldguide.org.nz/SoftwareEngineering.html#activity-fun-with-the-waterfall-process</p> <p>Class activity which implements the waterfall process.</p> <p>For this activity you will need a pack of cards and at least 6 people. You then need to divide the students into two groups of 3–4 people and place two chairs 30cm apart. The idea is to make a bridge out of cards.</p> <p>However, before you start, each group should write a specification about how they are going to tackle the problem. The other group then has to build the bridge to that specification. If it works, great, if not they have to improve the specification. This is an agile approach.</p> <p>Groups will have to keep developing ideas through iteration to get the best approach.</p> <p>You could assign a prize to the best specification. Extra points for diagrams!</p>	<p> Click here</p>
<p>Activity 2 – Project: Software processes http://www.csfieldguide.org.nz/SoftwareEngineering.html#project-software-processes</p> <p>This activity talks about how to prepare for having a visitor in to talk through software processes.</p> <p>Obviously this would require some pre-planning, but will bring this part of the course to life as the students might learn a bit about how software engineering really relates to different application development methodologies and whether there are any special things to take into consideration.</p> <p>This activity might also help forge links for potential apprenticeship routes, so is worth doing well.</p>	<p> Click here</p>
<p>Activity 3 – Algorithm development (CSInside) http://csi.dcs.gla.ac.uk/workshop-view.php?workshopID=12</p> <p>This activity is useful for introducing the idea that algorithms should be easy to understand and unambiguous. A presentation is provided and works with any class size or ability range.</p> <p>Students have to firstly follow a list of instructions for drawing a kite that are ambiguous, so they will probably all fail at drawing a kite. They then have to fix the instructions to be unambiguous. They can then design their own shape and develop instructions for that one.</p>	<p> Click here</p>



Thinking Contextually

Activities	Resources
<p>Activity 4 – Searching algorithms (CSUnplugged) http://csunplugged.org/searching-algorithms</p> <p>A great activity which covers Linear searches, Binary and Hashing using a game of battleships to illustrate the concepts. You will need to print out the worksheets provided on the website to run the activity.</p>	<p> Click here</p>
<p>Activity 5 – Sorting algorithms (CSUnplugged) http://csunplugged.org/sorting-algorithms</p> <p>A great activity which covers Insertion sort, Bubble sort and merge sort. Students will discover that some sorting algorithm methodologies produce much faster results than others.</p> <p>This activity requires quite a bit of preparation since you will need 8 containers that are the same size but contain different weights. It is also important that you cannot see what's inside the containers. Film canisters are perfect for this. You will also need some balance scales, or some can be constructed out of Lego or wood etc.</p>	<p> Click here</p>





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:
<http://www.ocr.org.uk/qualifications/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

OCR customer contact centre

General qualifications

Telephone 01223 553998

Facsimile 01223 552627

Email general.qualifications@ocr.org.uk



For staff training purposes and as part of our quality assurance programme your call may be recorded or monitored.

©OCR 2015 Oxford Cambridge and RSA Examinations is a Company Limited by Guarantee, Registered in England.
Registered office 1 Hills Road, Cambridge CB1 2EU. Registered company number 3484466. OCR is an exempt charity.