OCR
Oxford Cambridge and RSA

Computer Science

AS/A Level

H046/H446

Our vision

‘A high quality computing education equips pupils to understand and change the world through computational thinking. It develops and requires logical thinking and precision, it combines creativity and rigour: pupils apply underlying principles to understand real-world systems.’

Overview

The new specification covers all aspects requested by leading HE providers, industry leaders and both CAS and the BCS.

COMPUTING AT SCHOOL
EDUCATE · ENGAGE · ENCOURAGE
In collaboration with BCS, The Chartered Institute for IT

bcs The Chartered Institute for IT
Enabling the information society

OCR
Oxford Cambridge and RSA

Stakeholders

- Sheffield Hallam University
- University of Cambridge
- De Montfort University
- The University of Birmingham
- Brooke Weston Academy
- Stourport High School
- Downing College
- Kings College London
- Cardiff Metropolitan University
- Birmingham City University
- The Higher Education Academy
- University of Roehampton
- The Open University
- Stafford University
- Chalfonts Community College
- University of the West of England
- Living Networks
- Bright Learning Solutions
- BCS Academy of Computing
- COMPTIA UK
- ORACLE
- Microsoft
- E-skills UK
- NAACE
- Nesta
- Next Gen Skills
- The National Museum of Computing
- Toshiba Information Systems
- CISCO Systems
- TIGA
- Association of project Management

What has changed?

- Pretty much everything!
- Inline now with HE and Industry expectations
- Comparison of complexity of algorithms
- Number representation and bases
- Use any relevant language, Python, C++, Java etc
- Project more refined and more focussed on coding
- Higher weighting for skills
- Relevant and engaging

OCR
Oxford Cambridge and RSA

AS Level Computer Science

First teaching from 2015

Structure of AS Level

- Independent Qualification
- 2 papers
  1) Computing principles
     - 1 hour 15 minutes
     - 70 marks
  2) Algorithms and problem solving
     - 1 hour 15 minutes
     - 70 marks
- 50% weighting for each paper

Paper 1: Computing principles

- Traditionally marked and structured question paper
- Mix of question types:
  - short-answer,
  - longer-answer
  - levels of response mark scheme- type questions

- Characteristics of contemporary processors
- Software and software development
- Programming
- Exchanging data
- Data types, structures and algorithms
- Legal and ethical issues

Paper 1: Computing principles

- Components of a computer
  - Structure and function of the processor
  - Types of processor
  - Input output and storage
- Software and software development
  - Operating systems
  - Applications generation
  - Introduction to programming

Paper 1: Computing principles

- Exchanging data
  - Databases
  - Networks
  - Web technologies
- Data types, structures and algorithms
  - Data types
  - Data structures
  - Boolean algebra

Paper 1: Computing principles

- Legal, social, moral, cultural and ethical issues
  - Computer related laws
  - Ethical issues

Paper 2:
Algorithms and problem solving

- The principles of computational thinking
- Problem solving and programming
- Algorithms

- This paper contains a scenario based section with several questions exploring a single theme

Paper 2:
Algorithms and problem solving

- Elements of computational thinking
  - Thinking abstractly
  - Thinking ahead
  - Thinking procedurally
  - Thinking logically
- Problem solving
  - Programming techniques
  - Software development

Paper 2:
Algorithms and problem solving

- Algorithms
  - Analysis and design of algorithms
  - Standard algorithms
  - Suitability of algorithms

OCR
Oxford Cambridge and RSA

A Level Computer Science

First teaching 2015

A Level Structure

Content Overview	Assessment Overview	
• Characteristics of contemporary processors • Software and development • Exchanging Data • Data types structures and algorithms • Legal, moral and ethical issues.	Computer systems (01) 140 Marks 2 hours 30mins	**40%** of total A Level
• Elements of computational thinking • Problem solving and programming • Algorithms to solve problems and standard algorithms	Algorithms and programming (02*) 140 Marks 2 hours 30mins	**40%** of total A Level
• Analysis of the problem • Design of the solution • Developing the solution • Evaluation	Programming project (03 or 04*) 70 Marks Non-exam assessment	**20%** of total A Level

Structure

- 2 papers and 1 non-exam assessment
- Computer systems
- Algorithms and programming
  - Algorithms and programming has an extended scenario based section
- Each paper is 2 hours 30 minutes
- Each paper is 140 marks - 40% of total mark
- Programming project
  - 70 marks - 20% of total mark

Computer systems

- Characteristics of contemporary processors
- Software and development
- Exchanging data
- Data types structures and algorithms
- Legal, moral and ethical issues

Computer systems

- Characteristics of contemporary processors
  - Structure and function of the processor
  - Types of processor
  - Inputs, outputs and storage
- Software and software development
  - Systems software
  - Applications generation
  - Software development
  - Types of programming language

Computer systems

- Exchanging data
  - Compression, encryption and hashing
  - Databases
  - Networks
  - Web technologies
- Data types, structures and algorithms
  - Data types
  - Data structures
  - Boolean algebra

Computer systems

- Legal, moral and ethical issues
  - Computer related laws
  - Moral and ethical issues

Algorithms and programming

- Thinking abstractly
  - Thinking ahead
  - Thinking procedurally
  - Thinking logically
  - Thinking concurrently
- Problem solving and programming
  - Programming techniques
  - Computational methods

Algorithms and programming

- Algorithms
  - Analysis and design
  - Suitability
  - Measures to determine efficiency
  - Comparison of complexity
  - Standard algorithms

Programming project

- A coded solution to a problem:
- Analysis
  - Problem identification
  - Stakeholders
  - Research
  - Specify the proposed solution

Programming project

- Design
  - Decompose the problem
  - Describe the solution
  - Describe the approach to testing
- Develop
  - Iterative development
  - Testing to inform development

Programming project

- Evaluation
  - Testing to inform evaluation
  - Success of the solution
  - Describe the final product
  - Maintenance and development

Project Complexity Discussion

- What would make a good A-Level project?
- How do we help students choose an appropriate project?
- What have we looked for in the past?
- Complexity guide

Programming languages

- The project must be a coded solution, some languages are specified
  - Python
  - C family of languages (for example C# C+ etc.)
  - Java
  - Javascript
  - Visual Basic
  - PHP
  - Delphi/Lazarus
  - Swift
- Others by negotiation

Co-teachable

- The AS is designed to be co-teachable with the A level.
- The basic structure is similar but with extended content for A level
  - Content is broken at a sensible point to allow A Level students to return to these at a later stage.
- Co-Teaching Guide

Assessment

- We have used a best fit approach
- Grid of descriptors to refine the marking criteria and expectations
- Similar to old style of assessment
- Again, annotation on URS to support Moderator is important

Example of Marking Grid

AO 2.2 Analysis (maximum 10 marks)			
1–2 marks	**3–5 marks**	**6–8 marks**	**9–10 marks**
The candidate will have:			
• Identified some features that make the problem solvable by computational methods. • Identified suitable stakeholders for the project and described them and some of their requirements. • Identified some appropriate features to incorporate into their solution. • Identified some features of the proposed computational solution. • Identified some limitations of the proposed solution. • Identified some requirements for the solution. • Identified some success criteria for the proposed solution.	• Described the features that make the problem solvable by computational methods. • Identified suitable stakeholders for the project and described how they will make use of the proposed solution. • Researched the problem looking at existing solutions to similar problems identifying some appropriate features to incorporate into their solution. • Identified the essential features of the proposed computational solution. • Identified and described some limitations of the proposed solution. • Identified most requirements for the solution. • Identified some measurable success criteria for the proposed solution.	• Described the features that make the problem solvable by computational methods and why it is amenable to a computational approach. • Identified suitable stakeholders for the project and described them and how they will make use of the proposed solution and why it is appropriate to their needs. • Researched the problem in depth looking at existing solutions to similar problems identifying and describing suitable approaches based on this research. • Identified and described the essential features of the proposed computational solution. • Identified and explained any limitations of the proposed solution. • Specified the requirements for the solution including (as appropriate) any hardware and software requirements. • Identified measurable success criteria for the proposed solution.	• Described and justified the features that make the problem solvable by computational methods, explaining why it is amenable to a computational approach. • Identified suitable stakeholders for the project and described them explaining how they will make use of the proposed solution and why it is appropriate to their needs. • Researched the problem in depth looking at existing solutions to similar problems, identifying and justifying suitable approaches based on this research. • Identified the essential features of the proposed computational solution explaining these choices. • Identified and explained with justification any limitations of the proposed solution. • Specified and justified the requirements for the solution including (as appropriate) any hardware and software requirements. • Identified and justified measurable success criteria for the proposed solution.

0 marks = no response or no response worthy of credit.

Technical support

- Extensive appendix in the specification
- Identifies standards within exam papers for:
  - Pseudocode language
  - Boolean Loigic
  - ER Diagrams
  - HTML, JS, CSS, LMC, SQL
- This clearly delimits the knowledge students will need for this specification

Extensive set of resources

- Topic Exploration Packs
  - Learning, Activities and Key Points
- External resources
  - Endorsed resources for specification
- Delivery Guides
  - Support in Planning and Teaching content

Ocr.org.uk/gcsecomputerscience

Additional Support

- 3x subject specialists
- A-level MOOC – in development
- Exam creator - 2017
- Centre visits/support
- Regional Hubs via CAS
- KS4-5 transition guide
- KS5 – HE transition guide
- Curriculum planning guide
- Lesson Elements with resources
- CUP ebooks (some free)
- Hodder course text books
- 100 coding challenges
- Project complexity guide

Specimen Material

- AS and A Level Specimen Papers publically available
- Coursework will follow once first set of exams is complete (2017)
- New AS and A Level papers on **Interchange** early 2016
  - Allows for setting of unprepared mocks
- Exam Creator:
  - Content will build as qualification develops
  - Cross over with some of Old Spec stuff
  - Looking to develop more questions (12 months for a paper!)
- Project Work
  - Example of F454 game based one for old spec
  - Current sample material was chosen with new spec in mind

Supporting Marking

- New CPD Course in development
- Looks at assessing current F454 work with new mark scheme
- Identifies where key differences in mark scheme lie
- Provide initial guidance in supporting best practice for coursework delivery

activeresults

You can use it to:

- Review reports on performance:
  - individual candidates
  - cohorts of students
  - whole centre
- Analyse results at question/topic level
- Compare your centre with OCR national averages or similar OCR centres

It allows you to:

- Identify trends across the centre
- Gain additional insight to support decisions e.g. whether to apply for an enquiry about results
- Facilitate effective planning and delivery of courses
- Identify areas of the curriculum where students excel or struggle
- Help pinpoint strengths and weaknesses of students and teaching

New Professional Development Certification

- Open access through MOOC
- Comp Sci Certified course via university
  - Free Choice: Basic Certification
  - Paid for: university accreditation for Masters points
- Core modules
  - Development of SoW and delivery
  - Reflective Essay on success/learning
- 3 or 4 months to complete?

Keep in Touch

- OCR Website – ocr.org.uk/computerscience
- **@ocr_ict** - tweet/follow for resources
- CAS: we monitor CAS boards and attend Hub meetings
- Facebook: OCR Computer Science (A Level)
  - Please check messages – we require verification before accepting
- **Customer Contact Centre**
  - Tel: 01223 553998
  - Email: computerscience@ocr.org.uk

Questions?