Accredited

AS and ALEVEL Crossover Reference Guide

H046/H446

COMPUTER SCIENCE

September 2015



Itp://w



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 (<u>www.ocr.org.uk</u>) 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.

AS/A level content crossover reference

This resource highlights the differences between the section headings between AS and A Level and what content crosses over between AS and A Level.

| A2 | Section | AS | Section |
|-------|---|-------|---|
| 1.1.1 | Structure and function of the processor | 1.1.1 | Structure and function of the processor |
| 1.1.2 | Types of processor | 1.1.2 | Types of processor |
| 1.1.3 | Input, output and storage | 1.1.3 | Input, output and storage |
| 1.2.1 | Systems Software | 1.2.1 | Operating Systems |
| 1.2.2 | Applications Generation | 1.2.2 | Applications generation |
| 1.2.3 | Software Development | 1.2.3 | Introduction to programming |
| 1.2.4 | Types of Programming Language | 1.3.1 | Databases |
| 1.3.1 | Compression, Encryption and Hashing | 1.3.2 | Networks |
| 1.3.2 | Databases | 1.3.3 | Web Technologies |
| 1.3.3 | Networks | 1.4.1 | Data Types |
| 1.3.4 | Web Technologies | 1.4.2 | Data Structures |
| 1.4.1 | Data Types | 1.4.3 | Boolean Algebra |
| 1.4.2 | Data Structures | 1.5.1 | Computing related laws |
| 1.4.3 | Boolean Algebra | 1.5.2 | Ethical Issues |
| 1.5.1 | Computing related laws | 2.1.1 | Thinking abstractly |
| 1.5.2 | Moral and ethical Issues | 2.1.2 | Thinking ahead |
| 2.1.1 | Thinking abstractly | 2.1.3 | Thinking procedurally |
| 2.1.2 | Thinking ahead | 2.1.4 | Thinking logically |
| 2.1.3 | Thinking procedurally | 2.2.1 | Programming techniques |
| 2.1.4 | Thinking logically | 2.2.2 | Software Development |
| 2.1.5 | Thinking concurrently | 2.3.1 | Algorithms |
| 2.2.1 | Programming techniques | | |
| 2.2.2 | Computational methods | | |
| 2.3.1 | Algorithms | | |





Component 01 Computing Principles

BLUE text represents A Level only content within specification.

| AS and A Level | | AS only | | A Level only |
|-------------------|-------|---|-----|---|
| 1.1 | 1.1.1 | Structure and Function of the Processor | а | The Arithmetic and Logic Unit (ALU) |
| | | | | The Control Unit and Registers (Program Counter (PC), Accumulator(ACC), Memory Address Register (MAR), Memory Data Register(MDR), Current Instruction Register (CIR) |
| | | | | Buses (data, address, control) |
| | | | | How all these relate to assembly programs |
| | | | b | The fetch-decode-execute cycle |
| | | | | It's effect on registers |
| | | | с | The factors affecting the performance of CPU (clock speed, number of cores, cache) |
| | | | d | The use of pipelining in a processor to improve efficiency |
| | | | e/d | Von Neumann, Harvard and contemporary architecture |
| | 1.1.2 | Types of Processor | a | The differences between, and uses of, CISC and RISC processors |
| | | | b | GPUs and their uses (including those not related to graphics) |
| | | | c/b | Multicore and Parallel systems |
| | 1.1.3 | Input, output and storage | а | How different input, output and storage devices can be applied as a solution to different problems |
| | | | b | The uses of magnetic, flash and optical storage devices |
| | | | с | RAM and ROM |
| | | | d | Virtual Storage |
| 1.2 | 1.2.1 | Systems Software/Operating Systems | а | The need for, function and purpose of Operating Systems |
| | | | b | Memory Management (paging, segmentation and virtual memory) |
| | | | с | Interrupts, their role and Interrupt Service Routines (ISR), their role within the fetch-decode-execute cycle |
| | | | d | Scheduling: round robin, first come first served, multi-level feedback queues, shortest job first, shortest remaining time |
| | | | e | Types of Operating Systems: distributed, embedded, multi- tasking, multi-user, real-time |
| | | | f | BIOS |
| | | | g | Device drivers |
| | | | h | Virtual machines, any instance where software is used to take on the function of a machine, including executing intermediate code or running an operating system within another |
| | 1.2.2 | Applications Generation | а | The nature of applications, justifying suitable applications for a specific purpose |
| | | | b | Utilities |
| | | | с | Open source v closed source |
| | | | d | Translators: interpreters, compilers, assemblers |





| AS and A Level | | AS only | | A Level Only |
|-------------------|-------|-------------------------------------|-----|--|
| | | | e | Stages of compilation: lexical analysis, syntax analysis, code generation, optimisation |
| | | | f | Linkers, loaders, uses of libraries |
| | 1.2.3 | Software Development | а | Understand the waterfall cycle, agile methodologies, extreme programming, the spiral model, rapid application development |
| | 2.2.2 | Software Development | b | The relative merits and drawbacks of different methodologies and when they might be used |
| | | | с | Writing and following algorithms |
| | | | d | Different test strategies: black and white box testing, alpha and beta testing |
| | | | e | Test programs that solve problems using suitable test data and end user feedback, justify a test strategy for a given situation |
| | 1.2.4 | Types of Programming Language | а | Need for and characteristics of a variety of programming paradigms |
| | 1.2.3 | Introduction to Programming | а | Procedural programming techniques: program flow, variables and constants, procedures and functions, arithmetic/Boolean/ assignment operators, string handling, file handling |
| | | | b | Procedural languages |
| | | | c/b | Assembly Language (including following and writing simple programs with the Little Man Computer Instruction set) |
| | | | d | Modes of addressing memory: immediate, direct, indirect and indexed |
| | | | е | Object-oriented languages with an understanding of classes, objects, methods, attributes, inheritance, encapsulation, polymorphism |
| 1.3 | 1.3.1 | Compression, Encryption, Hashing | a/b | Lossy v Lossless compression |
| | 1.3.3 | Web Technologies | b | Run length encoding and dictionary coding for lossless compression |
| | | | с | Symmetric and asymmetric encryption |
| | | | d | Different uses of hashing |
| | 1.3.2 | Databases | а | Relational database: flat file, primary key, foreign key, secondary key, entity relationship modeling, normalisation and indexing |
| | 1.3.1 | Databases | b | Methods of capturing, selecting, managing and exchanging data |
| | | | с | Normalisation to 3NF |
| | | | d | SQL interpret and modify |
| | | | е | Referential integrity |
| | | | f | Transaction processing, ACID (Atomicity, Consistency, Isolation, Durability), record locking, redundancy |
| | 1.3.3 | Networks | a | Characteristics of networks, the importance of protocols and standards |
| | 1.3.2 | Networks | b | The internet structure: the TCP/IP stack, DNS, Protocol Layering, LANs and WANs, Packet and Circuit switching |
| | ļ | | с | Network security and threats, use of firewals/proxies/encryption |
| | ļ | | d | Network hardware |
| | ļ | | e | Client-server and peer-to-peer |
| | 1.3.4 | Web Technologies | а | HTML, CSS, Javascript |
| | 1.3.3 | Web Technologies | b | Search Engine Indexing |





| AS and A Level | | AS only | | A Level Only |
|-------------------|-------|------------------------------------|-----|---|
| | | | с | PageRank Algorithm |
| | | | d | Server and client side processing |
| 1.4 | 1.4.1 | Data Types | a | Primitive data types (integer, real/floating point, character, string, Boolean) |
| | | | b | Represent positive integers in binary |
| | | | с | Use of sign-and-magnitude, and two's complement to represent negative binary numbers |
| | | | d | Addition and subtraction of binary integers |
| | | | e | Represent positive integers in hexadecimal |
| | ĺ | | f | Convert positive integers between Binary, Hexadecimal and Denary |
| | | | g | Representation and normalisation of floating point numbers in binary |
| | | | h | Floating point arithmetic: positive and negative numbers, addition and subtraction |
| | | | i | Bitwise manipulation and masks: shifts, combining with AND/OR/XOR |
| | | | j/h | How character sets (ASCII, UNICODE) are used to represent text |
| | 1.4.2 | Data Structure | a | Arrays - up to 3D - records, lists, tuples |
| | | | b | Structures to store data: linked-list, graph (directed, undirected), stack, queue, tree, binary-search tree, hash table |
| | | | с | How to create, traverse, add data to and remove data from the data structures mentioned above (either using arrays and procedural programming or OO approach) |
| | 1.4.3 | Boolean Algebra | a | Define problems using Boolean logic |
| | | | b | Manipulate Boolean expressiosn, including the use of Karnaugh maps to simplify Boolean expressions |
| | | | с | Use the following rules to derive or simply statements in Boolean algebra: De Morgan's Laws, distribution, association, commutation, double negation |
| | | | d/c | Use logic gate diagrams and truth tables |
| | | | e | The logic associated with D type flip flops, half and full adders |
| | 1.5.1 | Computing related legislation | a | The Data Protection Act 1998 |
| | | | b | The Computer Misuse Act 1990 |
| | | | с | The Copyright, Deisng and Patents Act 1988 |
| | | | d | The Regulation of Investigatory paowers Act 2000 |
| | 1.5.2 | Ethical, moral and cultural issues | a | The individual (moral), social (ethical) and cultural opportunities and risks of digital technology Computers in the workforce Automated decision making Artificial Intelligence |
| | | | | Environmental Effects Censorship and the internet Monitor behaviour Analyse Personal Information Piracy and offensive communications Layout, colour paradigms and character sets |



Component 02 Algorithms and Problem solving

BLUE text represents A Level only content within specification.

Light Green text represents AS only content within specification.

| AS and A Level | | AS only | | A Level Only | | |
|-------------------|--|------------------------|-------------------|---|--|--|
| 2.1 Elemen | 2.1 Elements of Computational Thinking | | | | | |
| | 2.1.1 | Thinking Abstractly | а | Identify the components of a problem | | |
| | | | b | The need for abstraction | | |
| | | | с | The differences between an abstraction and reality | | |
| | | | d | Devise an abstract model for a variety of situation | | |
| | 2.1.2 | Thinking ahead | а | Identify the inputs and outputs for a given situation | | |
| | | | b | Determine the preconditions for devising a solution to a problem | | |
| | | | с | The nature, benefits and drawbacks of caching | | |
| | | | d/c | The need for reusable program components | | |
| | 2.1.3 | Thinking procedurally | а | Identify the components of a problem | | |
| | | | b | Identify the components of a solution to a problem | | |
| | | | с | Determine the order of the steps needed to solve a problem | | |
| | | | d | Identify sub-procedures necessary to solve a problem | | |
| | 2.1.4 | Thinking logically | а | Identify the points in a solution where a decision has to be taken | | |
| | | | b | Determine the logical conditions that affect the outcome of a decision | | |
| | | | с | Determine how decisions affect flow through a program | | |
| | 2.1.5 | Thinking concurrently | а | Determine the parts of a problem that can be tackled at the same time | | |
| | | | b | Outline the benefits and trade offs that might results from concurrent processing in a particular situation | | |
| 2.2 Probler | n solving | and programming | | | | |
| | 2.2.1 | Programming Techniques | а | Programming Constructs - sequence, iteration, branching | | |
| | | | b | Recursion, how it can be used, and compares to an iterative approach | | |
| | | | c/b | Global and local variables | | |
| | | | <mark>d/</mark> c | Modularity, functions and procedures, parameter passing by value and by reference | | |
| | | | e/d | Use of an IDE to develop/debug a program | | |
| | | | f | Uses of object oriented techniques | | |
| | 2.2.2 | Computational Methods | а | Features that make a problem solvable by computational methods | | |
| | | | b | Problem recognition | | |
| | | | С | Problem decomposition | | |
| | | | d | Use of divide and conquer | | |
| | | | е | Use of abstraction | | |





| | | | f | Learners should apply their knowledge of: |
|-------------|-------|------------|-----|---|
| | | | | Visualisation to solve problems |
| | | | | Data mining |
| | | | | Heuristics |
| | | | | Performance modelling |
| | | | | Pipelining |
| | | | | Visualisation to solve problems |
| 2.3 Algorit | nms | | | |
| | 2.3.1 | Algorithms | a | Analysis and design of algorithms for a given situation |
| | | | b | The suitability of different algorithms for a given task and data |
| | | | | set, in terms of execution time and space |
| | | | С | Measures and methods to determine the efficiency of different |
| | | | | exponential and logarithmic complexity) |
| | | | с | Implement bubble sort, insertion sort |
| | | | d | Comparisons of the complexity of algorithms |
| | | | d | Implement binary and linear search |
| | | | e | Algorithms (represenging, adding to) for the main data |
| | | | | order) and breadth first traversal of trees) |
| | | | f/b | Standard algorithms (Bubble sort, insertion sort, merge sort, |
| | | | | quick sort, Dijkstr's shortest path algorithm, A* algorithm, binary search and linear search) |





Component 03 or 04

BLUE text represents A Level only content within specification.

| Analysis of the problem (10 marks) | | | | | |
|------------------------------------|------------|----------------------------------|----------|--|--|
| AS and A Level | | AS only | | A Level Only | |
| | 3.1.1 | Problem identification | а | Describe and justify the features that make the problem solvable by computational methods | |
| | | | b | Explain why the problem is amenable to a computational approach | |
| | 3.1.2 | Stakeholders | а | Identify and describe those who will have an interest in the solution explaining how the solution is appropriate to their needs (this may be named individuals, groups or persona that describes the target end user) | |
| | 3.1.3 | Research the problem | а | Research the problem and solutions to similar problems and justify suitable approaches to the solution | |
| | | | b | Describe the essential features of a computational solution explaining those choices | |
| | | | С | Explain the limitations of the proposed solution | |
| | 3.1.4 | Specify the proposed solution | a | Specify and justify the solution requirements including hardware and software configuration (if appropriate) | |
| | | | b | Identify and Justify measurable success criteria for the proposed solution | |
| 3.2 Design | of the sol | ution (15 marks) | | | |
| | 3.2.1 | Decompose the problem | а | Break down the problem into saller parts suitable for computational solutions justifying any decisions made | |
| | 3.2.2 | Describe the solution | a | Explain and justify the structure of the solution | |
| | | | b | Describe the parts of the solution using algorithms justifying how these algorithms form a complete solution to the problem | |
| | | | С | Describe usability features to be included in the solution | |
| | | | d | Identify key variables/data structures/classes justifying choices and any necessary validation | |
| | 3.2.3 | Describe the approach to testing | а | Identify the test data to be used during the iterative development and post developmnet phases and justify the choice of this data | |
| 3.3 Develo | ping the s | olution (25 marks) | | | |
| | 3.3.1 | Iterative development process | а | Provide annotated evidence of each stage of the iterative development process justifying any decisions made | |
| | | | b | Provide annotated evidence of prototype solutions justifyin any decisions made | |
| | 3.3.2 | Testing to inform development | a | Provide annotated evidence of testing at each stage justifying the reason for the test | |
| | | | b | Provide annotated evidence of any remedial actions taken justifying the decision made | |
| 3.3.3 Testin | g to infor | m evaluation- may also be 3.4. | 1 Testir | ng to inform evaluation | |
| 3.4 Evaluat | ion (20 m | arks) | | | |
| | 3.4.1. | | a | Provide annoated evidence of testing the solution for robustness at the end of the development process | |



| | b | Provide annoated evidence of usability testing (user feedback) |
|-------|---|---|
| 3.4.2 | a | Use the test evidence from the development and post development process to evaluate the solution against the success criteria from the analysis |
| 3.4.3 | a | Provide annotated evidence of the usability features from the design, commenting on their effectiveness |
| 3.4.4 | a | Discuss the maintainability of the solution |
| | b | Discuss potential further development of the solution |



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

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. OCR acknowledges the use of the following content:

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 2014 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.