

Accredited

GCSE 2012

Computing

Teachers' Handbook

J275

Version 1

September 2012

CONTENTS

Introduction	3
Teacher guidance	4
Resources	10
Other forms of support	12
Frequently asked questions	14

INTRODUCTION

OCR's GCSE in Computing entered first teaching in September 2010.

We have improved the quality of our GCSEs for teachers and students alike. We've made improvements in two key areas: updated and relevant content and a focus on developing students' personal, learning and thinking skills.

In addition and in response to reforms announced by the Government and in response to Ofqual mandated changes to GCSEs, unitised assessment of this qualification is being replaced by linear assessment from September 2012. This means that candidates commencing a two year course from September 2012 will take all of their GCSE units at the end of the course in June 2014.

The main changes are:

- controlled assessment and examinations will be summative
- examinations provide opportunity for extended writing and more varied question types
- all GCSEs will meet the requirements of the Equality Act
-

OCR offers a range of support materials, developed following extensive research and consultation with teachers. We've designed them to save you time when preparing for the specification and to support you while teaching them.

It is important to make the point that this Teacher's Handbook plays a secondary role to the specifications themselves. The GCSE Computing specification is the document on which assessment is based: it specifies what content and skills need to be covered. At all times therefore, the Teacher's Handbook should be read in conjunction with the Specification. If clarification on a particular point is sought, then that clarification must be found in the Specification itself.

TEACHER GUIDANCE

UNIT A451: COMPUTER SYSTEMS AND PROGRAMMING

Relationship to other units

There is a certain amount of correspondence between the content of unit A451 and the other two units in the specification.

It is, perhaps, helpful to use the following analogy. We would reasonably expect professionals or experts within the field of computing to be able to demonstrate their expertise by recalling and applying a certain body of knowledge immediately, when called to do so. There are, however, certain activities within their expertise that we would not expect them to be able to demonstrate immediately. We would expect them to be able to perform certain practical tasks, access reference materials, carry out research and be aware of recent developments within computing, in response to a specific need. We would also expect them to be able to develop software, but appreciate that this would happen over a period of time.

If we scale this analogy down to the level of a GCSE candidate, the rationale for the three-unit model becomes clearer. Unit A451 focuses on the knowledge, understanding and skills which the candidates, being our “experts” at this level, can reasonably be expected to demonstrate immediately, independently and under examination conditions. The other “expert” knowledge and, especially, skills are more fairly assessed by controlled continuous assessment.

The specification for unit A451 therefore sets out precisely that body of knowledge from which the examination will be based. In reality, controlled assessment tasks for unit A452 will all be rooted in content within the specification topics 2.1.1 to 2.1.6 of unit A451. In the controlled assessment, just as in the real world, our “expert” will be expected to carry out research and practical tasks to update and further their understanding of these areas. Similarly, the candidates’ ability to program is assessed mainly in unit A453. However, we would expect our “experts” to be able to demonstrate some programming knowledge and skills under examination conditions. Topic 2.1.7 of unit A451 outlines precisely what is expected. It is not anticipated, however, that this would be taught

separately from unit A453. In fact, it is by studying unit A453 that students will acquire the knowledge and skills required to answer the questions which fall under this section.

Suggested Delivery

While we appreciate that each centre will choose the mode of delivery which best suits the centre and the candidates, it may be useful to consider the following model, in view of the close integration of the three units. Over the duration of the course, the candidates study A451 and A453 in parallel developing both their understanding of the theory and their programming skills. Within the study of A451, after the relevant topic for the chosen A452 assessment, the A452 task is given. This will enable candidates to further consolidate their knowledge of the A451 content in preparation for their examination, as well as give them a strong basis for the controlled assessment. Where the course is delivered over two years, centres may even be able to do more than one A452 controlled assessment task over the duration of the course as they fall within the topics. (The candidate will be awarded the highest mark achieved).

Meanwhile, as the candidates are learning to program in preparation for doing the controlled assessment for A453, the content of topic 2.1.7 of unit A451 should be reinforced using exercises, quizzes and sample/past examination questions. This will help put the questions into context.

Specific guidance on individual topics

TOPIC 2.1.1 – FUNDAMENTALS OF COMPUTER SYSTEMS

Note the model of a computer system described. The specification moves away from a narrow definition of a computer as the traditional desktop or notebook computer, and includes a wide variety of processor-controlled devices which operate under this model such as mobile phones, video games consoles and other consumer electronic devices. We want candidates to be aware of the pervasiveness and relevance of computer technology in the modern world. Examination questions will focus especially on devices which would reasonably be within the experience of the average teenager in the UK.

TOPIC 2.1.2 – COMPUTING HARDWARE

It is highly recommended that, where possible, candidates should be physically handling, installing and testing

the hardware devices studied, or at least, watching demonstrations of them. They should also research current specifications of personal computer systems on the market.

The understanding of the operation of the CPU required at this level is limited to understanding that instructions are fetched from memory and executed in the processor and that these instructions usually involve receiving data (from memory/input), performing calculations and returning the results (to memory/output). The Little Man Computer model is excellent for teaching this, and there are several simulations of this available on the Internet, one of the most suitable at this level being found at <http://www.cs.ru.nl/~erikpoll/III/lmc/>.

Binary logic is included so that candidates understand that this is the fundamental basis of the circuitry in the processor. This provides a rationale for studying the binary representation of data in topic 2.1.3. There are obvious parallels with their understanding of Boolean operators in their programming. "Simple logic diagrams" in examination questions will usually be limited to 3 AND, or NOT gates. Computer simulations of logic circuits or LED, logic gates and switch kits currently used in teaching GCSE Electronics will be invaluable.

TOPIC 2.1.3 SOFTWARE

Candidates should be given the opportunity to have firsthand experience of the software in this topic, especially system software. They should ideally have the opportunity to perform a range of system maintenance tasks using utilities. Where possible, experiencing and comparing a number of operating systems and applications of the same kind will help candidates understand the fundamental features of the software types.

The term "open source" is used in contrast to proprietary software and refers to all forms of free (as in speech) open source software. Candidates and centres should avoid the assumption that such software is necessarily free of charge, whereas proprietary software has to be bought.

TOPIC 2.1.4 REPRESENTATION OF DATA IN COMPUTER SYSTEMS

In this topic, candidates should gain an understanding of how different data types are internally represented in binary. Only the currently most common usage of the term byte (i.e. 8 bits) is used. Candidates are expected to know that a kilobyte = 1024 bytes, a megabyte = 1024

kilobytes etc. (For some candidates, this may mean using a calculator in the examination, which is permitted). On bitmap images, working with the Netpbm portable bitmap file format (.PBM, .PGM and .PPM) will allow candidates to create images and view their representation in a text editor such as notepad.

On sound, candidates need to understand that sound is essentially a wave and that sampling can be used to generate a series of numbers, coded in binary, from which the wave can be reconstructed. A detailed understanding of the nature of sound is not necessary. The sound = a wave concept can be easily demonstrated using several sound recording software packages.

TOPIC 2.1.5 DATABASES

It is anticipated that most students will learn the content of this topic using database management software such as Microsoft Access. However, candidates should be encouraged to develop a broader view of what a database is, and appreciate how fundamental they are to many applications they use regularly (Web 2.0 applications, supermarket point of sale systems etc).

Candidates are not expected to formally normalise a database into third normal form from a list of attributes. They should understand why it is necessary to store entities in separate table (e.g. to avoid repetition, to ease data entry and updating, to prevent inconsistency etc...). They should also understand the need to use the primary key from one table as the foreign key in another to link them. Examples in examination questions will usually be limited to 2 tables/entities and will, in all cases, never exceed 3.

No specific format for formulating queries is expected, allowing centres to choose the tools most suitable to teach the concepts for their candidates. Examples of queries in examination questions will be given in plain (if somewhat structured) English, and candidates answers will only be considered for the correct use of criteria on logical operators as specified. As long as these are clear in the candidates answer, they can answer in any format they have learnt.

TOPIC 2.1.6 COMPUTER COMMUNICATIONS AND NETWORKING

Candidates should be familiar with networks from their centre's networks and their use of the Internet. Many will also have networks in their homes, often wireless networks and may have handheld devices

such as game consoles, mp3 players or mobile phones which can connect to a network. This experience should be used as a basis to study this unit, giving the candidates a deeper understanding of networks than they would be expected to have just from normal use. However, if it is practical for the centre to set demonstrations where candidates actually network stand-alone a number of stand-alone computers.

Many networking devices are multi-purpose, eg a modem, hub, router and wireless access point all rolled into one. Candidates should be taught to understand the fundamental role of each of the device types and to recognise the different functions within the one device.

TOPIC 2.1.7 PROGRAMMING

As previously discussed, this topic tests the candidates' understanding of fundamental programming concepts which are common to most imperative languages. It is intended that most of the candidates' understanding of the concepts here will have been learnt from their experiences of programming. The ability to program (and hence their understanding of this topic) will normally take time and centres are advised not to consider this topic as being a separate theoretical topic.

Centres can choose which language to use in teaching the concepts here (as they do for A453) as long as it will allow candidates to understand and explain the concepts below. Note that where an algorithm is used in a question, it will be presented in a generic BASIC/Pascal – like pseudocode form which should be understandable to most candidates. Examples of such code can be seen in OCR's current examination papers for GCE A-Level Computing unit F452. Unlike the A-Level however, algorithms in this unit will usually be shorter and will not need line numbers for reference.

Similarly, when candidates write code in answer to a question, only the logic of the solution will be considered. Candidates will be free to answer in the language they have studied, in generic pseudocode, structured English or in diagrammatic form, as long as the logic is clear.

UNIT A452: PRACTICAL INVESTIGATION

The material covered in the investigations is intended to extend that which is itemised in the rest of the specification, so that the study of the subject remains interesting and challenging. This unit is intended to be largely practical based but the candidates' work will need to be presented in a clear and logical manner so that assessment is possible. Background research and preparation are required for the successful completion of this assessment. Some preparatory investigations may be teacher guided although the candidate is expected to show as much independence and initiative as possible. Once the assignment itself is under way, the amount of teacher guidance should be limited. For example, if coding is required, the student may be taught relevant techniques in advance, but should carry out the actual assessed tasks unaided.

The candidate will produce a report which should be structured according to the headings given in the particular assignment that has been chosen. The report will be assessed as a whole under the criteria headings given.

TOPIC 2.2.1 PRACTICAL ACTIVITY

The student should approach the tasks in a methodical way, so some evidence of planning is expected. For example, the hardware and software that are required should be stated with reasons given. Thought should be given to the order in which sub tasks are to be carried out so that important stages are not omitted. The assessment will normally involve the acquisition of new skills and these should be identified in advance as far as possible. Background research may be necessary and if this is relevant, this should be planned and documented.

In order to score marks in the higher bands, the work needs to be clearly presented and showing evidence of independent work.

TOPIC 2.2.2 EFFECTIVENESS AND EFFICIENCY

How this is shown will depend greatly upon the nature of the task chosen. For example, if there is an element of coding, then the algorithms used should ideally be the minimal required to achieve the objectives. If data files are involved, these should be organised in a logical and clear manner such that any future maintenance would be assisted. This criterion also allows a judgement to be made as to whether the solution to each task is completely successful or only partially so.

A report that scores highly in this section will demonstrate a complete solution to all the tasks and will show conciseness both in its execution and the presentation of evidence of success.

TOPIC 2.2.3 TECHNICAL UNDERSTANDING

This criterion focuses on whether the candidate clearly understands the details and processes involved in the tasks. A good candidate will be confident and fluent in the use of technical terminology and show in depth understanding of the central and peripheral issues involved. The best reports will contain significant detail so that the reader is in no doubt of the candidate's competence.

TOPIC 2.2.4 TESTING, EVALUATION AND CONCLUSIONS

The nature of the testing will vary considerably according to the tasks chosen. The best reports will show that testing has been as complete as is appropriate to the situation. They will be logically set out and show that the testing has been planned. They will communicate the purpose and results of the testing with clarity and economy of language.

As with the programming unit, A453, detailed means sufficiently detailed to clearly explain, identify or describe that component and not that the section is complete in every respect with no omissions. The middle mark bands where detail is not required imply that there are obvious omissions and the description does not cover all the necessary aspects. In the lower mark bands there will be significant lack of detail obscuring the descriptions.

UNIT A453: PROGRAMMING PROJECT

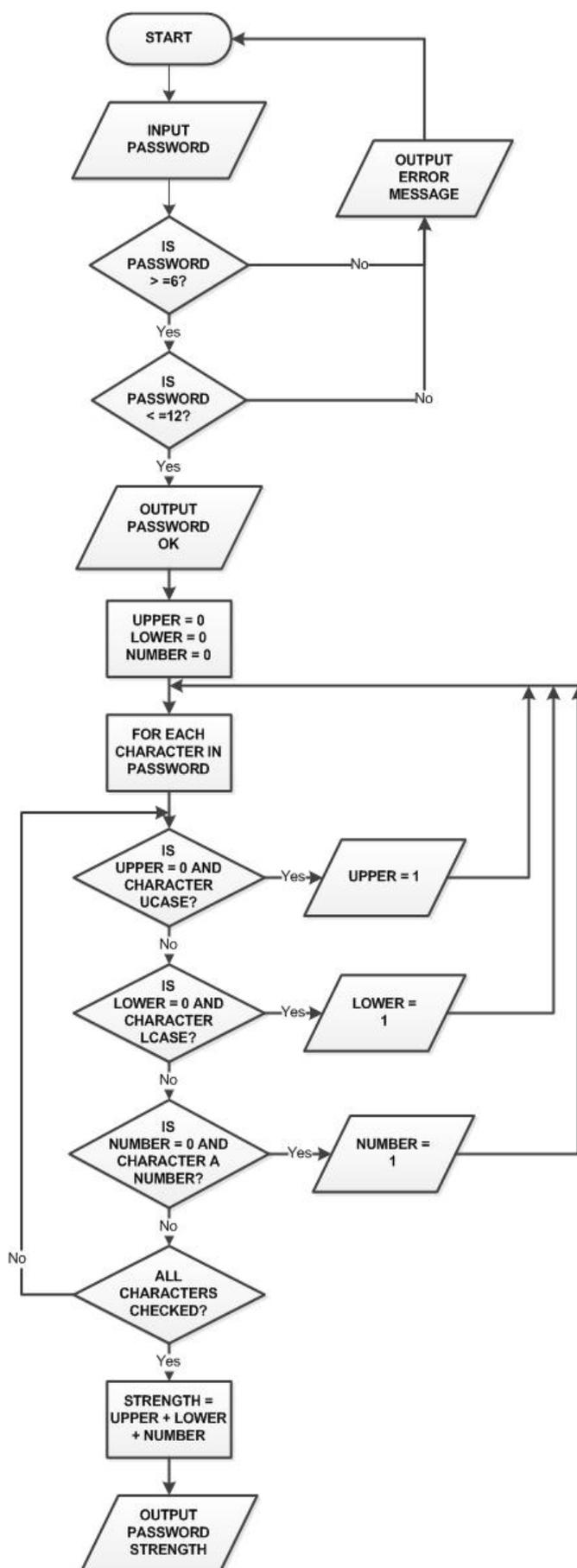
The basic programming concepts required for this unit should be covered alongside the theory unit at the appropriate time. Students will need to be able to relate the practical activity to the theoretical concepts being developed throughout unit A451. This unit is intended to assess the students' ability to use a range of programming techniques effectively and efficiently to produce a solution to a set of problems.

The basic scenarios will be explained for candidates so they need only identify what they are being asked to do and design a suitable solution to this before getting to grips with the coding of their solution. These designs are intended to demonstrate their ability to define a solution in an appropriate format using suitable algorithms. It is the logic of these algorithms that will be assessed not any specific format. It is important that candidates show how these algorithms relate to the identified tasks, how they form a complete solution and they plan to test that they have achieved the desired outcomes.

For example the password task in the sample assessment material requires several stages:

1. Input a password
2. Is the password between 6 and 12 characters long?
 - No; reject and return to stage 1
 - YES; output message and carry on
3. Check each character of the password in turn
 - Is this character upper case? If yes flag that upper case is included
 - Is this character lower case? If yes flag that lower case is included
 - Is this character a number? If yes flag that number is included
4. If three flags set then the password is STRONG
5. If two flags set then the passwords is MEDIUM
6. If one flag set then the password is WEAK

This outline explains what is required but now it needs to be put into an algorithmic form, for example a flowchart.



The pseudo code based on this algorithm:

```

REPEAT
INPUT the password
len=length of password
IF len <6 OR len >12 THEN
PRINT suitable error message
UNTIL len >=6 and <=12
PRINT password OK
Initialise upper, lower and number to 0
FOR i = 1 TO len
IF MID$(password, i, 1) is upper AND upper =0
THEN upper =1
ELSE
IF MID$(password, i, 1) is lower AND lower =0 THEN
lower =1
ELSE
IF MID$(password, i, 1) is number AND number =0
THEN number =1
NEXT i
strength = upper+lower+number
CASE
strength = 1 then PRINT "WEAK"
strength = 2 then PRINT "MEDIUM"
strength = 3 then PRINT "STRONG"
  
```

Having designed the algorithm and perhaps the pseudo code it is possible to check that this will solve the problem as stated and produce the code for the solution. The quality and care taken over the design will, in many cases, influence the quality and efficiency of the solution. Candidates need to show that they have checked the algorithms do produce the desired results and that they have considered any validation they will have to include before proceeding to code the solution.

For example will they be rejecting passwords that contain other characters besides alphanumeric ones and, if so, how?

The candidate should now proceed to code their solution testing individual elements of the code as they proceed, for example they should code the 'enter password and check for length between 6 and 12' section before considering the following parts of the problem. This process should be recorded, preferably in a diary and preferably in an electronic diary such as a web log. Evidence of testing for each of the coded elements is essential evidence for the development of the solution. For the higher range of marks candidates also need to use meaningful variable names, commenting, well organised program structures and suitable formatting to make the solution easy to follow.

The specification credits efficient solutions and what constitutes an efficient solution is a subjective matter. The efficiency of the solution will depend upon several factors including the choice of programming language. A system that works consistently and without any significant delays and solves the problem is effective. What makes a solution efficient are the logical processes in the design stage and the use of programming techniques appropriately to solve the problem. For example in the password task the use of the CASE statement for outputting the password strength is more efficient than using a series of IF THEN statements, but checking every character in the password rather than simply searching for a single existence of upper, lower and number may be considered less efficient. The code that would follow directly from the pseudo code above would probably be regarded as efficient and score in the top range of the marks for that section, but not necessarily the top mark in that range. The specification also credits the use of a range of techniques; this should not be read as every technique must be used. The tasks will provide scope for many techniques to be used, but the credit is for using these appropriately to solve the problem and not simply for using as many as possible.

The final part of the task is to test the solution with suitable test data and evaluate the final product. Other students should be called upon to take part in this process providing feedback on the solutions to the 'developer'. The solutions need to be tested against the original requirements and the identified success criteria from the design stage. Candidates should evaluate their solutions using their and others evidence from testing to show how the final solution compares to the original requirements for the tasks. The candidate's evaluation their system should indicate how they have tested the against the success criteria and include evidence to support their conclusions. Evidence provided can take several forms and this need not be a written report but the quality of the candidate's communication will be assessed throughout this section of their report.

RESOURCES

There are a huge number of resources on the web and these, of course, are changing all the time. In particular, teachers are reminded that groups such as CAS (Computers at School) and the American CSTA (Computer Science Teachers Association) <http://www.csta.acm.org/> are active in putting members in touch with up to the minute and exciting resources to back up the teaching of what is perhaps the most vibrant subject available to students.

Raspberry Pi:

The official Raspberry Pi website has forums and updates on developments. It is also a great place to see what people have been creating with the Raspberry Pi.

www.raspberrypi.org

We have been working in collaboration with Raspberry Pi and with leading practitioners to create resources to support the use of the Raspberry Pi in the classroom. You will find tutorials, a series of Classroom Challenges, a Resources Link detailing suitable external resources relating to the Raspberry Pi, plus additional items on our website.

www.ocr.org.uk/raspberrypi

Hodder Dynamic Learning:

Hodder are the official publishing partners for OCR and have developed a dynamic learning website to support the GCSE Computing specification. It contains a range of resources covering all aspects of the specification.

<http://www.dynamic-learning.co.uk>

Susan Robson has produced a text book covering the theory elements of this course and it is available as a pdf site licence, a monochrome printed text book or a full colour printed text book from:

<http://www.lulu.com/spotlight/susanjrobson>

theteacher.info has a range of resources dedicated to GCSE computing including a text book, a website, tests and self-marking tests.

<http://www.theteacher.info/index.php/gcse-computing>

GCSEComputing.org.uk is a website containing a wide range of resources dedicated to this specification including coverage of the theory and controlled assessment units.

<http://www.gcsecomputing.org.uk/>

CS4Fn:

Free magazine and website from Queen Mary College, London.

<http://www.cs4fn.org>

Industry publications:

There are many industry magazines which put computing into a work related context. As well as many articles and features on contemporary issues, it is useful to look at the jobs sections to see what skills are actually in demand at the moment.

<http://www.computerweekly.com/Home/>

http://s0.2mdn.net/1651284/ctg_HDS_welcome

<http://www.computing.co.uk/>

Computing at schools group:

Lots of support including teaching units for programming skills and other resources.

<http://www.computingatschool.org.uk>

Little Man Computer:

Just one of several online working demonstrations of how memory and the processor interact.

<http://www.atkinson.yorku.ca/~sychen/research/LMC/LittleMan.html>

Background reading for the more able students:

<http://www.eecs.qmul.ac.uk/~pc/research/education/puzzles/reading/>

<http://courses.cs.vt.edu/~csonline/index.html>

A visit to the **Bletchley Park Museum** always provides a stimulating background to the history and importance of computing.

<http://www.bletchleypark.org.uk/>

BBC BASIC for windows:

Easy to use programming language.

<http://bbcbasic.co.uk>

Computer Science Unplugged:

Set of exercises from Canterbury University in New Zealand.

<http://csunplugged.com/>

D F Stermole website:

Resources to support programming in various languages including Pascal, Java C etc plus an excellent introduction to key programming techniques in the Turing pages.

<http://www.dfstermole.net/>

Dick Baldwin:

Programming tutorials for various languages including some excellent SCRATCH tutorials.

<http://www.dickbaldwin.com/>

FreePascal:

Programming language

<http://www.freepascal.org/>

GameMaker:

<http://www.yoyogames.com/gamemaker>

Greenfoot:

Environment for introducing object oriented programming

<http://www.greenfoot.org/index.html>

Proanimate:

An interactive flowchart based visual problem solving tool and code generator. It is aimed at the basics of programming, problem solving and code reading skills.

<http://www.comp.glam.ac.uk/pages/staff/asscott/proanimate/>

Raptor:

Visual algorithm based application that allows students to create flowcharts that they can run and test.

http://www.usafa.af.mil/df/dfcs/bios/mcc_html/raptor.cfm

RoboMind:

Programmable robot environment

<http://www.robomind.net/en/index.html>

Scratch:

Great resource from MIT for introducing programming that uses colour coded snap together blocks to create sequences of instructions. A good starting point for developing the logical programming processes not hindered by the syntax barrier

<http://scratch.mit.edu/>

Python:

Python is a programming language that lets you work more quickly and integrate your systems more effectively. Python is an easy to learn programming environment.

www.python.com

For **further resources** to support teaching:

<http://info.scratch.mit.edu/Educators>

Small Basic:

A simple 3rd generation BASIC

<http://msdn.microsoft.com/en-us/devlabs/cc950524.aspx>

Visual Basic.net:

Standard free visual .net environment programming language and support.

<http://www.microsoft.com/express/Windows/>

Visual Basic Express books:

McGrath, M. Visual Basic Express in Easy Steps (2006) ISBN: 9781840783292

This is an excellent easy to follow introduction to the language.

Ford, J. Microsoft Visual Basic 2008 Express Programming for the Absolute Beginner (2009) ISBN: 978-1598639001

This is a more detailed book for those who want to go beyond the absolute basics of the language.

OTHER FORMS OF SUPPORT

In order to help you implement the new Entry Level Computing specification effectively, OCR offers a comprehensive package of support. This includes:

PUBLISHED RESOURCES

OCR offers centres a wealth of quality published support with a fantastic choice of 'Official Publisher Partner' and 'Approved Publication' resources, all endorsed by OCR for use with OCR specifications.

PUBLISHER PARTNERS

OCR works in close collaboration with three Publisher Partners; Hodder Education, Heinemann and Oxford University Press (OUP) to ensure centres have access to:

- Better published support, available when you need it, tailored to OCR specifications
- Quality resources produced in consultation with OCR Subject teams, which are linked to OCR's teacher support materials
- More resources for specifications with lower candidate entries
- Materials that are subject to a thorough quality assurance process to achieve endorsement

Hodder Education is the publisher partner for OCR GCSE Computing.



Hodder Education has produced the following resources for OCR GCSE Computing.

Dynamic Learning Website. Authors: George Rouse, Agneau Belanyek, Sean O'Byrne.

A stand-alone Dynamic Learning subscription website,

designed to be used by students and teachers in class and at home. It incorporates a lesson builder, search facility, VLE integration and electronic resources to support the course aims and objectives and includes automatically-marked interactive assessments.

Book: O'Byrne, S and Rouse, G. OCR Computing for GCSE (2012), ISBN: 978-1444177794

APPROVED PUBLICATIONS

OCR still endorses other publisher materials, which undergo a thorough quality assurance process to achieve endorsement. By offering a choice of endorsed materials, centres can be assured of quality support for all OCR qualifications.



ENDORSEMENT

OCR endorses a range of publisher materials to provide quality support for centres delivering its qualifications. You can be confident that materials branded with OCR's "Official Publishing Partner" or "Approved publication" logos have undergone a thorough quality assurance process to achieve endorsement. All responsibility for the content of the publisher's materials rests with the publisher.

These endorsements do not mean that the materials are the only suitable resources available or necessary to achieve an OCR qualification. Any resource lists which are produced by OCR shall include a range of appropriate texts.

OCR PROFESSIONAL DEVELOPMENT

The OCR Professional Development Programme offers more accessible and more cost effective training, with the same valued content that you expect from us.

At OCR, we are constantly looking for ways in which we can improve the support we offer to teachers. Most recently we have been considering the increasing challenges that schools face in releasing teachers

for INSET, and how OCR can make its professional development programme more accessible and convenient for all.

From September 2012, our new improved programme will include:

- FREE online professional development units available when and where you want them
- FREE live web broadcasts of professional development events
- FREE face to face training for GCSE controlled Assessment and GCE coursework
- a series of 'not to be missed' premier professional development events

For more information, please email training@ocr.org.uk or visit www.ocr.org.uk/training

OCR SOCIAL

Visit our social media site <http://www.social.ocr.org.uk>. By registering you will have free access to a dedicated platform where teachers can engage with each other - and OCR - to share best practice, offer guidance and access a range of support materials produced by other teachers; such as lesson plans, presentations, videos and links to other helpful sites.

INTERCHANGE

OCR Interchange has been developed to help you carry out day to day administration functions online, quickly and easily. The site allows you to register and enter candidates online. In addition, you can gain immediate and free access to candidate information at your convenience. Sign up at <https://interchange.ocr.org.uk>

FREQUENTLY ASKED QUESTIONS

CONTROLLED ASSESSMENT

Are new tasks set every year with a new mark scheme?

Yes and No, tasks will be set and be available for a number of examination sessions. New tasks will be produced over time and we may retire older tasks during the life of the specification. The mark scheme is generic and is the same for all tasks.'

Can I just choose electronic or postal submission once I have the work from the candidate and use the most appropriate method?

No, you must decide when entering the candidates which method you will use and you must send work by that method to the moderators.

Do all of my candidates have to do the same set of tasks?

No, you can choose from the available tasks to suit the needs and interests of individual candidates.

Do the candidates have to demonstrate every attribute in a mark band in order to achieve the relevant mark?

No, it is a best fit approach.

Do we have to do specific topic each year from the pre-release materials?

No, OCR will be issuing two topics initially with more to follow later; you are free to choose from any of the currently available topics.

What formats can we use? Can we assume the moderators have access to any necessary programs?

OCR publishes a list of acceptable formats and moderators will have access to these but you cannot assume the moderators will have access to any specific software and you must ensure common formats are used.

Will tasks be retired while my candidates are working on them and I will have to start again?

No, we will give plenty of notice if we decide to withdraw a task and this will be well before any candidates have started to work on them.

A452

Can candidates work together on the assignments?

Yes, they are expected to, but the reports must be their own and the part that each candidate plays must be clearly documented. Cooperation will probably be limited because each candidate is expected to demonstrate personal ability to perform the practical tasks.

Can the candidates simply research these topics on the internet and produce a report?

No, they should supplement the findings from their practical investigation with research from the internet, where appropriate, but they are meant to spend a significant proportion of their time in practical activity for this unit.

Can the work be submitted electronically or must it be a typed report?

The work can be submitted electronically or via the postal system. The work may take the form of a typed report but there must be evidence of practical activity and this may be more easily presented in other ways than a typed report. If submitted electronically, then any of the approved formats will be acceptable. If submitted by post, then these are still acceptable if sent on a virus checked medium.

Must the candidates carry out the practical investigations personally or can they research the answers from other sources such as the internet?

Candidates must carry out practical work and demonstrate that they have done so.

The specimen scenario contains terms that are not in the specification. To what extent are the candidates expected to use of terms beyond the material given in the specification?

They should look up all the terms that are in the scenario. This will lead to other terms that will probably be relevant in their responses. The candidates should display a secure understanding of the scenario by the correct use of technical terms obtained from their researches so it is expected that there will be terms and concepts that are not specified in the specification.

What sort of tasks do you envisage we will see over the next few years?

We plan to develop a set of tasks that extends elements of each major part of the specification and hope these will be used to extend candidates' knowledge of a wide range of topics.

Will candidates be penalised if their secondary sources are all web based?

No, most information is available online these days and to insist on paper resources is unrealistic.

A453

Can candidates use additional modules to produce solutions, for example those available within VB.net?

Yes, using additional modules is acceptable, though in many cases not necessary but the use of such modules may make for more attractive and functional results. This extra functionality will not necessarily attract extra marks.

Can I select the most suitable subtasks for my candidates from the options available?

No. The sets of tasks must be seen as a single item and you may not pick and choose from within these. The sets of tasks have been designed to cover as many aspects of the specification as feasible and form a single assessment.

Can the candidates use any method to solve the problems for example a spreadsheet or a database?

No, this is a coding exercise and tasks are set in order to test computer programming skills. If the question involves storing and retrieving data for example candidates are expected to produce a coded solution.

Do the candidates have to utilise all of the programming techniques to score full marks?

No, the techniques may not all be appropriate and different tasks, different approaches, different programming languages will make some techniques unsuitable. The effective and efficient use of a range of appropriate techniques from the specification is what we are looking for.

What programming language must we use?

You may use any suitable programming language to solve the problems. In some cases a recommendation will be made to ensure the tasks are not overly difficult but in general any language will be fine.

Contact us

Keep up to date with the latest news by registering to receive e-alerts at www.ocr.org.uk/updates

Telephone 01223 553998

Facsimile 01223 552627

Email general.qualifications@ocr.org.uk

