GCE

# Computing

Advanced GCE **A2 H447**

Advanced Subsidiary GCE **AS H047**

# OCR Report to Centres

## June 2012

Oxford Cambridge and RSA Examinations

OCR (Oxford Cambridge and RSA) is a leading UK awarding body, providing a wide range of qualifications to meet the needs of candidates of all ages and abilities. OCR qualifications include AS/A Levels, Diplomas, GCSEs, OCR Nationals, Functional Skills, Key Skills, Entry Level qualifications, NVQs and vocational qualifications in areas such as IT, business, languages, teaching/training, administration and secretarial skills.

It is also responsible for developing new specifications to meet national requirements and the needs of students and teachers. OCR is a not-for-profit organisation; any surplus made is invested back into the establishment to help towards the development of qualifications and support, which keep pace with the changing needs of today's society.

This report on the Examination provides information on the performance of candidates which it is hoped will be useful to teachers in their preparation of candidates for future examinations.  It is intended to be constructive and informative and to promote better understanding of the specification content, of the operation of the scheme of assessment and of the application of assessment criteria.

Reports should be read in conjunction with the published question papers and mark schemes for the Examination.

OCR will not enter into any discussion or correspondence in connection with this report.

# CONTENTS

## Advanced GCE Computing (H447)

## Advanced Subsidiary GCE Computing (H047)

## OCR REPORT TO CENTRES

# Overview

There was less evidence in this session of candidates being entered for papers for which it was obvious they were not suited or had not done adequate preparation. The majority of candidates demonstrated an appropriate level of understanding while some scripts and projects were a pleasure to mark or moderate. The vast majority of candidates did themselves and their centres justice and I would like to thank, on behalf of the whole examining team, all the teachers who prepare these candidates year after year. We hope that all candidates have attained an outcome commensurate with their abilities and their efforts.

It is apparent that centres are largely realising that the content of F451 is very large and wide ranging and that to try to complete it with most candidates before the January session of exams is unreasonable, particularly given the very ‘dry’ nature of much of the content. It is designed to be taught alongside the material in F452 and as such it means that the two modules are likely to be examined at the same time during the June session.

Teachers are reminded that the final mark schemes for all the papers are available from the OCR website.

# F451 Computer Fundamentals

**General Comments:**

The paper worked well giving all candidates the opportunity to earn marks while providing challenges to all. Each question elicited the full range of marks from zero to maximum from at least some of the cohort, although the distribution of the marks was different according to the difficulty of the subject matter.

It is noticeable that phrases from previous papers are being used by many of the candidates to answer questions. There is nothing wrong with this, per se, as it is understood that there are a limited number of ways to word the answer to a question. However, when the wording in a response matches that used in a previous mark scheme to a similar, but different, question suspiscions arise that candidates are learning mark scheme answers. This was particularly evident in question 5 where the rolling mill and the computer controlling the thickness of the glass sheets made a re-appearance from a previous paper in some scripts. The mark schemes provide a very powerful resource for teachers to use to educate themselves as to the requirements of the paper and to share with candidates when they think it appropriate, but they should not be learned for their own sake.

The proportion of candidates who simply did not have the ability to have sensibly entered the examination continues to fall and while there will inevitably be candidates who are disappointed with their results we hope that the result, whatever it may be, is a true reflection of their efforts. Indeed some scripts were excellent showing a clear understanding of the concepts and an ability to handle the skills required.

There was no evidence of any candidate having suffered from any trouble completing the examination in the time provided.

**Individual questions:**

**Question 1**

**a** Most candidates scored well although some candidates failed to define software accurately.

**b** Some candidates had already given some of the expected responses here to the previous question. That did not matter if the response was repeated here.

**c** This was a good discriminator question. Most candidates typically gave enough evidence to score one mark for each part of the question, but it was typically only the more able who were able to score full credit.

**Question 2**

**a** This question was reasonably well answered. Many candidates, however, ignored the splitting up of the question and gave what amounted to a free flowing response across the two dotties. The question was marked allowing for such cross fertilisation of ideas so the candidates who did this were not penalised. Candidates should, however, be reminded to follow the instruction in the question.

**b** The requirement to list four features of the feasibility study proved surprisingly onerous for many candidates. The two parts of the question were intended to elicit the two types of answer: simple statements of features and then expansions to describe what the feature meant in a real application. In reality the two parts were marked together with comments in (ii) being used to support very weakly stated features in (i). Many candidates were unable to get past simple statements like ‘Is it legal?’ giving an expansion of ‘Is it against the law?’ At this level it would be reasonable to assume that having identified a feature the candidate could have said something about the Data Protection Act or some other act like

Copyright protection. Most candidates did not realise that budgetary questions are different to long term financial issues like profitability issues. Both were good points which could have been made.

**Question 3**

**a** There were six items which were being asked about in this question. There were the three named parts of the computer and the three busses. The three parts of the computer were named and should not hold any problems for those who have completed the course. Centres are advised that this is an area of the specification where the candidates in general are very weak. The busses suffered from candidates imbuing them with rather more power than they actually have. Most candidates had busses making decisions and controlling the parts of the computer. Typical words used with the busses were 'controls', 'sends', 'decides' and the most common of all 'stores'. All of these show a very hazy grasp of the concepts involved. A few candidates described other busses. While these were not expected because they are not named in the syllabus, they are perfectly acceptable responses to the question and were treated as such, as were responses where the registers took centre stage. These were not expected but are sensible features to consider in such a question, especially if a candidate was retaking this module having been taught the F453 content.

**b** This proved to be a relatively easy question. The first part was well answered. Some of the spelling attempts were questionable but we credited most, even that RAM was 'viotalic', the understanding was there. There are still a few candidates offering BIOS as being stored in ROM. Once again, it should be clarified that parts of the BIOS are stored in ROM, but other parts are not. If the candidate is specific then obviously they would be given the mark, but I have not seen any that have been specific, they simply say BIOS. As the whole reason for the BIOS is that it is user defined, it becomes impossible to have it all stored on ROM. As for the non-alterable parts, strictly that is not stored on ROM either in order to allow for upgrades in the software as and when it becomes necessary. There were a few candidates who stated that the Operating System would be stored on ROM. While this is true for some devices like some games machines, it is not true for computers.

**Question 4**

**a** The first part was well understood and executed. The BCD, however, was not so well understood and should perhaps be an area for centres to work on for future series as this is quite a straight forward concept.

**b** Surprisingly, the answers provided by candidates to the use of two's complement and the required subtraction were very good. It was good to see most candidates take note of the words in the question which told them to show their working. The majority provided a well annotated solution which was easy to follow, a marked improvement on previous questions about binary arithmetic.

**Question 5**

**a** Most candidates were able to score full marks here.

**b** Good answers from most candidates, although some gave generic answers rather than trying to fit them to the example application.

**c** The three parts of the expert system did not trouble most candidates, but the way in which they are used, did. It would have been nice to have the responses linked more obviously to the valve testing application but this was rare, being largely restricted to the most able candidates. The use of an expert system should be looked at by most centres as it is not generally understood.

(For those that may be interested the systems described in this question are loosely based on a real application of valve testing called trevi-testing which includes all the features used in the question including the expert system used by engineers in the field.)

**Question 6**

**a** Some very good answers. Unfortunately, some confused ones as well. Most common error was the idea that a WAN has a lot of computers in it, followed closely by the need to connect the machines using Wi-Fi. The more technical question did not expect anything unusual, these are standard responses and a candidate at this level should have no difficulty describing two of them. Unfortunately they did, the router/modem being the only mark that some managed.

**b** The majority of candidates scored full marks.

**c(i)** Well answered with many candidates scoring full marks. Most common error was to use baud rate as one of their answers despite it being in the question.

**(ii)** This was recognised as a difficult question. In order to earn all the marks candidates had to identify the two types of data, explain what effect time sensitivity of the data had on that data and then describe two types of data fitting the parameters from the example given. Most candidates were unable to progress past the concept of video stalling when the bandwidth was too low.

**Question 7**

It was hoped that candidates could be a bit adventurous in describing recent technological advances and their effects on people. There were two parts to the question and both parts needed to be addressed in order to score well. Most common response was that washing machines now did the washing. It was normally left unclear what this was replacing, but the result was the same: answers that were unconvincing and disappointing.

**Question 8**

This was well answered with most candidates being able to score well on the last question on the paper. It is intentional to have accessible marks at the very end of the paper and candidates should not believe that just because it is at the end it is going to be ‘hard’.

**a** Most candidates can now explain the difference between backup and archive, although some still find the concept of archiving data difficult to explain. The backup procedure which should be followed by the secretary was well done with most scoring full marks. The only problem here was the candidates’ understanding of what was described in the question. It was deliberately meant to be important data but small scale. Many candidates did not realise this and talked about the sort of procedure which would be suited to a bank with incremental backups throughout the day and full scale backups at night ‘when the computer systems will not be working’.

**b** The question did not ask for input/output peripherals, rather, it wanted peripherals that could be used by the secretary. One of the tasks of the secretary, which was outlined in detail in part a was the need to store data, be it the primary data in the system or the backup copies of the data. Consequently the response to this part of the question cannot be considered to be a full answer without considering a storage device.

# F452 Programming Techniques and Logical Methods

**General Comments:**

Although this unit is tested by an examination, it aims to assess the candidates' knowledge, understanding and skills in programming. Centres are advised that these skills are best acquired through programming practise. All the questions in the examination are framed within the context of programs which should be similar to the programs that candidates have practised in preparation for the examination so that they can put themselves in the position of the programmer in the question. There was often evidence of candidates not being familiar with common programming problems in the questions and they appeared to be answering from stock answers, notes or even past mark schemes (which relate to different questions and are often not directly applicable) rather than addressing the problem at hand.

That said, a number of marks are available in the examination paper for defining key programming concepts and there continues to be evidence that even where candidates demonstrate their understanding of these concepts by applying them correctly to the context and to program extracts, they are often unable to formulate accurate and precise definitions for them. All key terms are clearly stated in the specification and candidates should learn these, as well as have a clear understanding of how each term relates to the programming language they are studying.

**Individual Questions:**

**Question 1**

**a** This was generally well answered, although fewer candidates obtained full marks than had been expected. Almost all candidates gained the marks which were available for including the items listed in the question on their design of an interface, but many did not give enough attention to present a clear, effective interface, taking into consideration that it is a touch screen interface. The use of dropdown lists, for example, was very frequent although this is not the most effective way to enter marks on a touch screen device.

**b** This was generally well answered with the majority of candidates gaining all six marks. The most common error was to give the number of points for a gymnast as an integer, where it has been stated that fractions of points may be deducted. Candidates are always given latitude in the names of data types to allow for variations between languages, but should note that when justifying the data type, they should provide more information than the name of the data type they have given. For example, while "text" is accepted as an alternative name for "string", candidates did not get both marks for stating that the name of the gymnast is "text, because it is a piece of text".

**c** This question was intended to discriminate between candidates, with most candidates gaining some marks and only the best candidates being able to give a full explanation and justification of the use of records in this case to gain full marks. While it worked as a discriminator to a large extent, for those candidates who knew what a record format is, there was a large proportion of candidates who simply did not answer the question, or confused this with file organisation, suggesting that this was an area that was not sufficiently studied by many candidates.

**d** Most candidates were able to identify the logic error and correct it in part (ii), and gained full marks. However, in part (i) many lost marks for a less than complete definition of what a logic error is in the first place – often due to the inability to formulate the concept precisely enough in their own words.

**e** Candidates fared better with defining the term “array” in part (i) which has appeared in a different context in a number of past papers. In part (ii) they were required to demonstrate an understanding of this definition in the context of the language they have learnt, by declaring an array of 6 real numbers called RawScores. Candidates gained credit for showing an awareness that the identifier, data type and size were relevant to this declaration, irrespective of any errors in syntax. Those who have studied languages with dynamic typing and/or variable sized arrays were still able to access these marks, for example Python users could have created a list of 6 items called RawScores where all items were initialised to 0.0 (i.e. RawScores = [0.0, 0.0, 0.0, 0.0, 0.0, 0.0]). However, examiners made allowances for candidates who used languages where specifying the size or data type of an array is not the normal practice. Answers to part (iii) were particularly disappointing given that the question asked for candidates to give the basic algorithms on a one dimension array listed in the specification (sum, maximum and minimum). Too many candidates lacked precision in their algorithms, giving a solution which was in broad terms not significantly more detailed than the description in the question. For example, many candidates assumed that because the highest and lowest scores were “eliminated” this meant they have to be removed from the array (rather than simply deducted/discounted from the array sum).

**Question 2**

**a** This was generally well answered, and it was pleasing to see the majority of candidates applying a general definition of RAD to this specific scenario to gain all four marks available.

**b** Despite the fact that the question was preceded by a detailed stem, indicating what the input requirements for the program were, too many candidates gave a general definition of a number of types of validation checks (often being poorly expressed or including errors) which did not refer to the data given. Centres should remind candidates that in a question such as this, candidates need to show that if they were to write the program, they would know precisely which validation cases to implement and/or test.

**c & d** Most candidates obtained full marks in (c) for following a flow chart correctly, a few did not get the last case. However, the number of candidates who produced a correct flow chart for the scenario given in (d) was disappointing. Candidates were expected to analyse the requirements given in the question to produce a flowchart which would satisfy it and for this reason, the requirements were not given “as an algorithm” but rather in the terms that would typically be supplied by an end-user. Too many candidates simply followed the sequence in the question, resulting in the absurd situation where their algorithm checks whether a student has a B or above in English after determining that the student does not have a C or above.

**e** Not many candidates showed a good enough understanding of the process of installation. Most candidates identified that files needed to be copied to the target machine (often being vague about what files needed to be copied).The question asks for the key processes in installation that allow the program to be run correctly on the target machine, and no credit was given for other events during a typical installation that candidates gave such as inserting the CD, accepting terms and conditions etc.

**Question 3**

**a** Part (i) was well answered with the majority of candidates gaining full marks for describing the difference between global and local variables. This is basic theory, however, part (ii) was intended to test the candidates' experience of this theory in their programming practise and not many candidates appeared to have encountered the situation of conflicting identifiers or to reasonably describe one of the two ways that languages would deal with this situation.

**b** Most candidates correctly identified the local and global variables in the code extract. Centres should remind candidates that in a question where they are asked to identify something in the code, their answer should include just the item needed (in this case the variable name). Giving anything more (such as the whole line, or a variable declaration) suggests that the candidate does not know which part constitutes the answer.

**c** In parts (i) and (ii) it was disappointing that many candidates did not understand operation precedence (i.e. BODMAS) and fewer recognised DIV as the operator for integer division. The operators MOD and DIV have different symbols in different languages and centres must ensure that candidates recognise the symbols given in the specification as these are the ones which will be used in questions and be able to relate these to the symbols in the language they have studied. As a result, very few candidates realised that there were two errors in the expression given. A third of candidates corrected the error in the order of precedence; there were several unsuccessful attempts to do this which still had brackets in the incorrect place (demonstrating again an unwillingness to check if a resultant calculation made any sense). Only a handful realised there was an error with the rounding up in cases where the TimeInMinutes was a multiple of 60.

**d** This saw many very adequate answers, but these were often unbalanced, with far more appreciation of the positive than the negative (lack of commenting being the only regular in appearance). Also many candidates tended not to give a precise explanation of how each programming technique mentioned helps to make the code more readable/maintainable, other than making bland comments such as "it makes it easier to read or understand." As a result, although there was a good spread of marks, the number of candidates obtaining a mark for a high level response was disappointing.

**Question 4**

**a** Many candidates appeared not to differentiate between the precise meaning of a "parameter" in programming, and the meaning of the term in common parlance – giving answers that went along the lines of "limits within which a program should work" and were thus unable to score well in parts (i) and (ii). Those who knew what a parameter is, often defined it in their own terms and lacked the precision to gain both marks in (i) but correctly identified the parameter in (ii). This is another case where it is helpful for candidates to have learnt the definitions of key programming concepts.

**b** Most candidates understood the difference between assignment and comparison but few gained full marks because they were not able to give the name of the type of operation as stated in the specification. A common error was to refer to the comparison as a "conditional" operator rather than a "relational" or "comparison" operator. Where candidates knew the correct term, some lost marks in the further explanation by using the term twice (e.g. "in line 05 it is an assignment operator as it assigns a value to FirstChar) where the expansion point is precisely for explaining the term in the context of the question, not repeating it.

**c** Candidates gave a variety of responses, probably relating to their varied experience of encountering this issue as part of their programming. It is essential that, in preparing for this examination, they have had experience of writing programs where they have to compare strings and use character code to character conversion operators, so that they can apply these skills in context.

**d** A majority of candidates defined recursion correctly (although there was a small number of candidates who omitted the question or confused recursion with iteration). Most of those candidates who had the right definition were also able to identify where recursion takes place within the code.

**e & f** Most candidates answered (e) correctly, but were less successful in (f). A small number of candidates thought that an empty string contains a space. The concept of an empty string is so fundamental that candidates should have a clear understanding of it. Candidates who were in the right area often failed to gain full marks because their explanation lacked clarity and precision.

**g** This was designed to be a discriminator and a true test of the concept of recursion. While one in ten candidates simply omitted this, most candidates who did attempt it were able to trace through the first call of the function. When they got to the second call, they were imprecise in indicating clearly that a second call was being made with new parameters, often suggesting that control just passes back to line 02. Consequently, few understood how to return to the original call to continue it to the end of the function, and give the return value. A diagram is a very effective way of communicating this, but this was seen in only a few cases.

# F453 Advanced Computing Theory

**General points:**

Some candidates had prepared for the examination very well. They were able to give detailed responses to a number of questions and showed a good understanding of the topics. Both the candidates and their teachers should be very proud of their achievements.
Unfortunately, there were other candidates who showed little or no knowledge of some topics and were under-prepared for an examination at this level. A large number also produced a confusing mixture of incorrect technical terms, poor spelling and, in some cases, almost illegible handwriting.
Basic advice should be given to all candidates:

- It sounds obvious, but "read the question" is the most important rule. There are further detailed comments about this below.
- Use technical terms correctly.
- For this unit, the majority of candidates who answer correctly do not require extra paper.

**Individual questions**

**Question 1**

**a(i)** "Describe the ***PURPOSE*** of scheduling" – only five words, but many candidates completely ignored all but the last. Worse, some described round-robin scheduling as their entire answer. Had they read (ii), it was obvious that this would not gain marks, quite apart from the fact that such a response took no notice of the fact that the question asked for the purpose.

**a(ii)** Some candidates answered well here. Others wrote about "chunks of time" or similar. Very few indicated that time slices are fractions of seconds.

**a(iii)** Generally well answered.

**b(i)** Fairly well answered.

**b(ii)** Many missed the point here. Common errors were to compare the priorities of interrupts, or to write that the job to be processed was needed by another job so had its priority increased.

**Question 2**

**a** This was answered fairly well, with most able to distinguish between the types of translator.

**b** Again, a number of candidates ignored the word "purpose" and only described intermediate code.

**Question 3**

**a** Technical terms were used extremely badly here, with complete confusion between job, task process, program, instruction… All these terms were seen and appeared to be used at random. Some candidates wrote about pipelining (not required by this Specification).

**b** Many attempted a comparison with CISC, which was not required. Common answers included "less instructions" and "simple instructions so easy to write programs". Many candidates did not appear to understand this topic.

**Question 4**

The stem of the question stated in bold type that "all working must be shown" but a number of candidates ignored this. Many answered well, though a worrying number of candidates did not understand the concept of mantissa and exponent even though similar questions have appeared in numerous past papers.

**a** Some candidates obtained -6 for the exponent, presumably using sign and magnitude notation despite what was written in the question.

**b** The majority converted 24 to pure binary correctly, though many were confused about where to move the binary point or what value the exponent should have.

**c** Some candidates failed to do a calculation here. In (i), many made the mistake of quoting the maximum number possible in 8 bit two's complement binary instead of working out the number possible in the notation described in the question. In (ii), a common mistake was to give the mantissa as the 5 bits 10001 without the leading zero required.

**Question 5**

**a** This was answered quite well by the majority.

**b** Also answered quite well. A common error in the algorithm was to report an error and then go on to try to read data which did not exist.

**c** The correct use of pointers did not seem to be understood by some candidates. Weak candidates tried to move all the data along the queue, then move the pointers. A few did not use pointers at all. The majority did gain good marks for this, though the most common error was to add all three data items before checking if there was sufficient space for the items.

**Question 6**

**(i)** Many candidates gave incomplete answers. Examples were vague and often failed to mention which parts were unique or inherited.

**(ii)** Most candidates seemed to know about this but were unable to give clear answers. Many confused the terms "attribute" and "operation".

**(iii)** Most candidates answered correctly.

**Question 7**

**a(i)** Answers were quite good, though many candidates failed to mention local variables.

**(ii)** Most candidates gained marks here. A few confused parameters with the values allowed by a range check.

**b** Some candidates used poor notation. Others ignored the instruction to write the definitions "in their simplest form".

**Question 8**

**(i)** Most candidates gained marks here.

**(ii)** The most common error was to refer to "the instruction part of the instruction" when the question had already stated that the instruction was ADD 45. A few confused the terms opcode and operand.

**(iii),(iv)** Many candidates answered well. A few made the question more difficult by attempting to make up their own examples, some of which were in binary, instead of referring to the example provided in the question.

**Question 9**

**a(i)** Many candidates gained marks here. A common error was to say there was "no data duplication".

**(ii)** Many candidates were unable to give an example of an appropriate application.

**b(i)** This is in the Specification – but many made wild guesses at the words represented by the abbreviation SQL.

**(ii)** Most gained marks, though few said that views of data allow the user to access data without the need for technical expertise.

**(iii)** Very few mentioned the name of the view.

**Question 10**

**a** Many candidates gave careless answers here. For example "ShootId in PHOTO" is a foreign key but "ShootId in PHOTOSHOOT" is a primary key: hence "ShootId" alone is not acceptable as an answer.

**b** This was answered quite well.

**c** The answer expected was a CUSTOMER entity, as CustomerId appeared in the question. However, a number of candidates seemed determined to answer a different question about resolving many-many relationships and included it here. This was acceptable, though it was a pity that many of these candidates did not understand how to use a composite primary key.

**d** Many candidates did not understand the concept of a secondary key or were unable to describe its use adequately. Few mentioned that it could be used for both sorting and searching.

# F454 Computing Project

Once again the standard of work submitted for this unit was largely encouraging. We have seen some excellent examples of coding in a variety of languages including a significant increase in Java, C++, Python and PHP. While Visual Basic still remains the most popular choice it is encouraging to see candidates producing code for mobile phone apps and demonstrating that they are thinking more about the choice of language for their chosen project.

The range of projects remains interesting with the usual data based projects, games, utilities and, increasingly, web based solutions using on-line databases and multiple languages.

It is important that by the end of the definition the reader should have a good idea of the nature of the problem which requires computerised solution and, in general terms, the type of input and output involved. All too often the definitions do not provide this basic information.

The investigation section remains the weakest element in the projects and candidates must think beyond the basic interview with an end user and research the project more thoroughly. Candidates need to look at all aspects of the problem area and gather suitable research to inform the designs, for example research similar situations, existing solutions to similar problems and the background to the problem. Candidates will not be able to score high marks in this section unless they complete a thorough investigation of the problem and should include things like:

- other available solutions (in detail) and their pros and cons.
- prototyping, iteration with the client, use of diagrams to show data flow, before and after.

Design work was generally good but where data flow diagrams and entity relationship diagrams are used they must be correctly drawn, with appropriate labels to indicate what data is flowing, and the nature of relationships respectively. More labelling would be helpful. There were a surprising number of candidates that produced lots of diagrams with few labels, and little if any explanation. Sadly, a lot of candidates included screen shots of the actual forms they had created rather than plans of what they intended to produce.

Algorithms were generally improved on last year with some decent use of pseudocode, but few could be considered to be complete enough for a different developer to use. There were a few commendable attempts at dry runs and this is to be encouraged.

Testing strategies are often confused with testing or simply test plans. For a test strategy we would expect to see a brief account about the range of testing techniques that are going to be used with some explanation or justification for the choices.

There were some excellent examples of software development with candidates showing how each section of the code was developed then tested in an iterative approach to reaching a solution. Some submissions were merely pages of code with little to link these to the process and test evidence. In some cases test evidence was difficult to relate to the code produced and failed to demonstrate that the program had been tested effectively. Proper testing should try to break the program, not simply confirm that it works if the right data is entered. A small number of candidates merely completed a test plan indicating success or failure without any evidence of the test being carried out, others provided video evidence of the tests being completed to leave the moderator in no doubt that the system had been tested thoroughly.

The documentation section requires good on-screen support, this needs to be evidenced and must be more than just a few helpful error messages. Some of the documentation submitted was little more than a series of screen shots and as such failed to meet the criteria for the higher bands in this section.

In the evaluation it is important candidates refer to the success criteria they identified in the design section and cross reference the test evidence from the development and testing sections with this to demonstrate the success, or otherwise, of their project.

While the vast majority of centres submitted work in a well-organised and timely manner we are still seeing late submissions with incomplete paperwork and disorganised folders of evidence. This makes moderation very difficult and time consuming and makes it more likely important evidence may not be located.  Centres providing well–organised samples with helpful teacher notes explaining how, where and why marks were allocated avoid such issues arising. Moderators are not re-marking they are assessing how the centre has marked the work.

We are now happy to accept work in electronic format and this represents a significant time and cost saving for centres. Centres are encouraged to submit the work on a suitable storage device rather than as a large printed document.

**OCR (Oxford Cambridge and RSA Examinations)**
**1 Hills Road**
**Cambridge**
**CB1 2EU**

**OCR Customer Contact Centre**

**Education and Learning**
Telephone: 01223 553998
Facsimile: 01223 552627
Email: general.qualifications@ocr.org.uk

**www.ocr.org.uk**

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

**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**

**OCR (Oxford Cambridge and RSA Examinations)**
**Head office**
**Telephone: 01223 552552**
**Facsimile: 01223 552553**