

IGCSE ICT : A Coursework Guide for candidates

Message to candidates: This document is in no way to be regarded as a replacement of the official Coursework Guide as published by London Examinations. It is intended to be a combination of a simplified explanation of that guide and a checklist of items to include if you intend to get an A* grade.

Firstly, you need to choose a good problem. There is no formal requirement for the problem to be based on a real situation, but it is usually to your advantage if it is.

Why? Well, if you invent the situation, or use friends or family for role play, you may miss out on using other people's knowledge and experience, e.g. if you decide to make a stock control project for a small business, someone who is really involved in stock control is likely to be able to make useful suggestions as to how such a system might work. They may also be willing to offer constructive criticism throughout the development of your system and to carry out user testing. On the other hand, if you choose a friend or family member to be your user, they may know no more about stock control than you do, in which case they are likely to go along with your ideas and tell you that you have developed a perfect solution! This is only natural: after all, your friends and family would not wish to upset you by telling you that you've got it all wrong.

Next, having got a good problem and a knowledgeable user lined up, you need to consider how you will set out your report.

The official Coursework Guide indicates that you should present your report in five sections corresponding to the Assessment Criteria. This is good advice and should be followed as closely as possible. Your centre or supervisor may supply you with a template which gives you section and sub-section headings. This may also be useful, but only if it covers all the sections from the Coursework Guide, preferably in the same order. If it doesn't, make sure that you add in the missing bits, in the right places.

The Coursework Guide does not give an upper limit on the project size, but it should be possible to get an A* grade within thirty pages. This figure does not take into account the need to supply plenty of drawings of your design, and print-outs of evidence for your testing. Designs should of course be included in the main body of your report, but where there are large numbers of screen designs, form layouts, etc you could have one example in the main body and the rest in an appendix; suitably referenced of course. Test evidence - normally screen shots - also takes up a lot of paper and could be placed in an appendix. You would need to have references from your test plan showing where the results are shown.

Overall you should not produce more than 100 pages, even with appendices. If you find yourself going over this limit, consider putting several designs or test results on a single page.

Now to the report in detail. The next sections act as check lists with some explanatory notes. If you cannot tick off most - if not all - of the items then you are unlikely to be doing yourself justice and you should try to correct the situation.

IDENTIFY

Have you identified a problem?

Have you written it down on the first page of your report?

Is there some background information?

Have you described what happens at present?

Have you got some discussion of what is wrong with the present system?

Have you stated what your user wants, in general terms?

i.e. what they want the new system to do.

Have you discussed your user's general requirements with them and produced a list of specific, desired outcomes?

Have you translated your user's requirements into at least three computer-related tasks that you can do?

Are your tasks testable?

If they are not, how are you going to prove that you have completed them?

If you write about general things such as being able to find a record in 30 seconds, reducing staff or getting rid of a filing cabinet, it is going to be very difficult to prove that you have done it.

Have you considered alternative solutions to the problem?

These could be manual or involve the use of different software.

Have you justified your choice of solution?

ANALYSE

Have you identified the hardware?

'What is available' and 'what is needed', if that is different. The use of a PC will be assumed. You only need to mention other hardware which is relevant to your project, such as a printer or scanner.

Have you identified the software?

'What is available' and 'what is needed', if that is different.

Have you identified where your data is coming from?

Have you explained how you will collect that data?

Have you explained how you will prepare the data for entry into the computer?

Have you explained your data entry methods?

Have you provided some examples of the raw data?

You will need this so that you can show how your data moves through your system from start to finish.

Have you made a data flow diagram?

This is not compulsory but it is usually a lot easier to explain data movement and processing with diagrams than with text. You do not have to use the correct data flow or system symbols: a simple block and line flowchart will be enough as long as it is clear.

Have you explained your backup and security methods?

This should cover both your working files and your final product. The methods should be workable and appropriate to what you are making. There is no point in writing a detailed description of a multi-generation file system unless you are going to use one and have a good reason for doing so.

Have you considered different output methods and justified your choices?

'Different' here means using the software that you have already decided on. Discussion of different software types should have been written in Identify. Different outputs might involve printouts and screen displays.

DESIGN

Have you actually stated what you are going to make?

Have you made an initial design which describes how you will solve the problem?

You need enough detail for your user to be able to take your design and make sensible comments about whether it is suitable for solving their problem.

Have you taken your user's comments into account and stated what changes you will make in order to accommodate the user's wishes?

Now for your Final Design. This needs to be detailed enough that someone else (who does not know either you or your user) could make the final product that you had planned.

Do you have sketches of input forms?

Do you have sketches of screens?

Do you have sketches of output reports?

Are all your sketches annotated?

Do you have system flowcharts?

Do you have data flow diagrams?

Do you have algorithms?

Do you have macro designs?

Do you have function and formulae designs?

Do you have data structures?

e.g. tables, files, folder arrangements.

Do you have outline program structures?

e.g. module lists, sub-routines.

Do you have alternatives for all of your design components?

Have you stated your choices from the alternatives and justified those choices?

You will not need all of the above for all types of project, but you will need a substantial proportion of them for any project.

Have you broken down your project into sub-tasks?

Are they sensible and manageable sub-tasks?

Take care that your list of sub-tasks does not just become a list of project stages. Or, even worse, a list of instructions for using a software package.

Do you have a test plan?

Have you given reasons for your tests?

Do your tests demonstrate that the project's objectives have been achieved?

Have you considered validations?

Tests must cover all aspects of your work, e.g. for a spreadsheet you would need to test all the calculations and macros. For a database you would need to show that you can search your data and display appropriate results. For a program you would need to work out all possible paths and show that they work.

Do you have test data and expected outcomes for your tests?

You will need a range of test data. It must cover the normal operation of your system, as well as generating error messages.

IMPLEMENT

Have you finished your design?

Yes? Well done, go on to the next bit.

NO? WHY NOT! Go back and do it now.

Have you shown development from your design?

You must make it clear how you have used your design to achieve your solution.

Do you have a sequence of build, test, analyse, modify?

No-one puts together a perfect project at their first attempt, so don't pretend that you have. It might give you an ego boost but it will cost you marks. Lots of them. Quite possibly a complete grade or even two grades. Better to be proud of a better grade than of submitting a 'perfect' project.

Have you explained what you have done?

All the markers will see is your report. They do not know you or your user and they will not see your software running, so make sure you tell them what you did.

Does your final product match your design?

You will have described some modifications already in this section, but check your solution against what you said you were going to do. If it does not match and you have not already mentioned the change, explain it now, before you move on to testing.

Did you make a test plan in Design?

Yes? Well done, go on to the next bit.

NO? WHY NOT! Well, you are just going to have to go back and pretend that you did. This is not the easy option, but it is too late to do anything else. Make sure the test plan refers to the original design, not the amended version. Make sure that the plan covers everything, not just a 'perfect' version.

Have you followed the plan?

There may be a few changes needed due to your amendments, but you can change the plan here as long as you explain why. In any event, you must do what you said you would do.

Have you shown all your test results?

ALL your results, do not be tempted to think that the markers will not bother looking at them all. If you have 50 tests you MUST have 50 results. If the screen shots are in an appendix, make sure that you cross-reference them. In the test section, say where the results are. In the results, say which test(s) they refer to.

Have you discussed the tests that gave the wrong results?

There are marks for fixing problems, so do not pretend that everything went perfectly if it didn't. Discuss any problems that you found and then describe how you put things right. Don't forget re-tests.

EVALUATE

Do you have some good, testable objectives from Identify?

Yes? *Excellent, now you can write a worthwhile evaluation.*

NO? *WHY NOT! Now you have really got a problem. How can you discuss the effectiveness of your solution if you cannot show that you have met the objectives. Go back to Identify and do a rewrite so that there is something that you can discuss here.*

Have you restated your objectives?

Can you show, with evidence, that you have met each one?

Evidence could be a short description and a page reference pointing to where the actual evidence is. If the proof is not in your write up you will need to go back and do some more work. You need evidence for all of the objectives. If you cannot prove that you have succeeded then you have not succeeded.

Have you got your user to write some comments on your final product?

Are the user's comments constructive and relevant?

Have you looked at weaknesses in your solution?

Are they different from the ones your user suggested?

You cannot get credit for simply restating things that your user has told you.

Have you described how you could fix those weaknesses?

If the answer is that you need to finish something that you said you were going to do in the first place, then you are going down the wrong path.

Have you looked at possible future developments?

Be realistic here, the word is possible. If you could not make the change or addition relatively easily and cheaply then look for something else to discuss. Do not go for generalised stuff either: make your suggestions appropriate to your project, not something that could be tacked onto the end of anybody's project.

Finally, have a look at your presentation.

Have you spell-checked the report?

Have you proofread it?

Be careful here. If you simply re-read the report, you are likely to see what you thought you wrote, rather than what is really there. Try reading the pages in reverse order.

Have you got a contents or an index?

Have you numbered all the pages?

Do the numbers fit with any page references you have given in the report?

Is the report suitably bound together?

Have you got your name, candidate number, centre number, subject and report title on the front cover?

A* grades can never be guaranteed for an IGCSE, but if you have followed the advice given in this guide you will have a better chance of reaching your full potential.