

O Level Computing Coursework Guide for candidates

This guide is not to be regarded as a replacement for the official Coursework Guide published by London Examinations. It is intended to be a simplified explanation of that guide and a checklist of items that need to be included if you are to obtain 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 problem, but it would be to your advantage if it is.

Why? Well, if you invent the problem or use friends or family for role play, you might miss the opportunity to gain from other people's knowledge and experience. For example, if you decide to do 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 about how a system could work. They might also be willing to offer constructive criticism throughout the development of your system and do some 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 agree 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, 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 might also be useful, but only if it covers all of the sections from the Coursework Guide, preferably in the same order. If it doesn't, make sure that you add in the missing topics, in the right places.

The Coursework Guide states that the report should be **no more than 30 sides of A4**. Some reports will take more than thirty pages if they require additional pages for drawings of your design together with printouts of evidence of 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 others in an appendix with suitable referencing so that they can easily be found. Test evidence, normally screen shots, also take up a lot of paper and could be placed in an appendix. You would need references from your test plan to where the results are shown. The only sure way to do this is to number all the pages in the report.

When completed you should not have produced more than 100 pages, including the appendices. If you find yourself going over this limit, consider putting several designs or test results on a single page.

Each part of the report has marks allocated for it. You cannot gain marks for parts of the report that are missing. Look carefully at the checklist that follows. If you cannot answer "yes" to most of the questions then you are unlikely to be doing yourself justice and you should try to make the report more complete.

Problem analysis and specification

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 said what you intend to do about it?

Have you stated in general terms what your user wants?

i.e. what your user wants the new system to do

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

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

Have you broken the computer related tasks into sub-tasks and described each of them?

Can you test them after they have been done?

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

Have you stated your success criteria?

These will probably be a combination of some of your sub-tasks and your user's specific, desired outcomes. They need to be testable and provable. If you write about general matters like 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.

Design and planning of solutions

Have you discussed the hardware?

What is available and what is needed.

Have you made a choice?

Have you justified your choice?

Have you discussed the software?

What is available and what is needed.

Have you made a choice?

Have you justified your choice?

Have you described how you will solve the problem?

The final design needs to have enough detail to enable someone else, who does not know you or your user, to produce the final product you have planned.

Have you actually stated what you are going to make?

Do you have sketches of input forms?

Do you have sketches of screens?

Do you have sketches of output reports?

Are your sketches all 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.

Do you have a test plan?

Have you given reasons for your tests?

Tests must cover all aspects of your work e.g. for a spreadsheet you would need to test all of the calculations and macros. For a database you would need to show that you could search your data and display appropriate results. For a program you would need to follow all possible paths and show that they work. Most importantly, tests must demonstrate that you have met the user's requirements and the success criteria.

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

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

Have you considered validations?

Implementation of design.

Have you finished your design?

YES? Well done, continue.

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 that the examiners 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.

Testing of the solution.

Did you make a test plan in Design?

YES? Well done, continue.

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 examiners will not look 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 overcoming 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.

Evaluation of the solution.

Do you have some good, testable Success Criteria from Analyse?

YES? Excellent, now you can write a worthwhile evaluation.

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

Have you restated your Success Criteria or 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 criteria or objectives. If you cannot prove that you have succeeded then you have not succeeded.

Have you discussed how the hardware performed?

Have you discussed how the software performed?

Have you discussed how the Human Machine Interface performed?

Have you looked at weaknesses in your solution?

Have you described how you could fix those weaknesses?

If the answer to that one 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?

Do not simply re-read it. You wrote the report and know what it says, so that is what you are likely to see when you read it, even if it isn't really there. Read the pages backwards or in a random order, that way you will have more chance of spotting the mistakes.

Have you included a contents or an index page?

Have you numbered all of the pages?

Do the numbers agree with all the page references that 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 at O Level but if you have followed the advice given in this guide you will have a better chance of reaching your full potential.