Appendix 1: Instructions and guidance for candidates

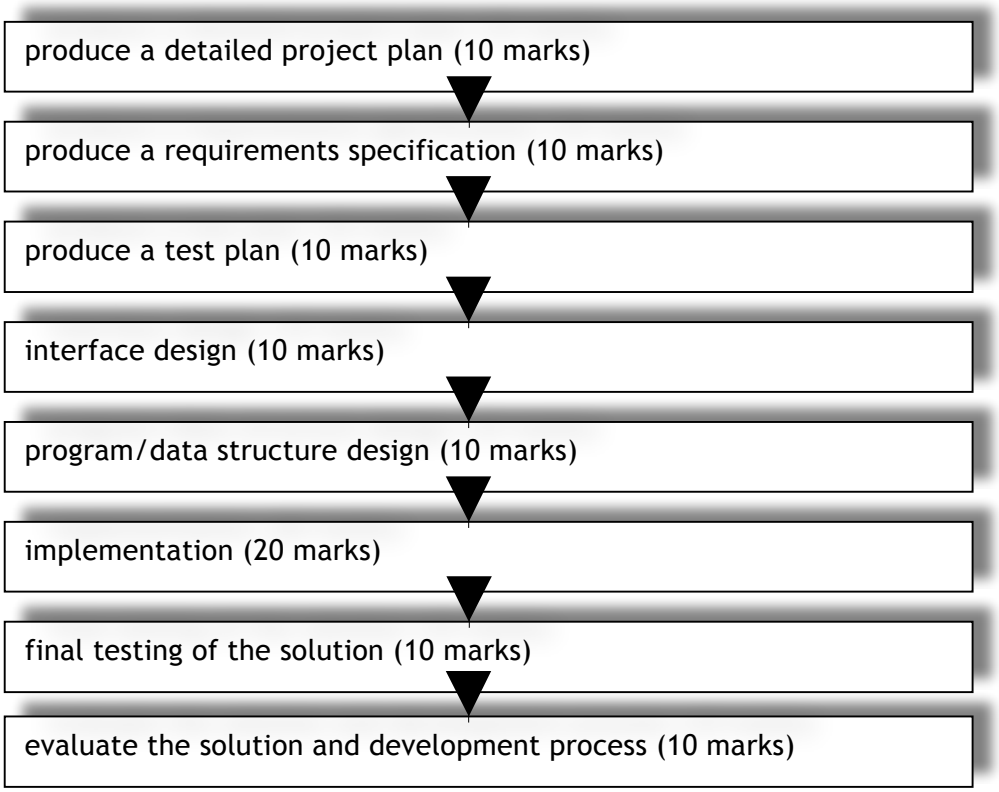
Computing Science project

These instructions and guidance apply to the project for Advanced Higher Computing Science.

This project is worth 90 marks out of the total of 150 marks. This is 60% of the overall marks for the Course assessment. The Course will be graded A–D.

Your assessor will let you know how the assessment will be carried out and any required conditions for doing it.

The project will be based on a computing science problem agreed with your assessor. You will be required to research and analyse a problem, then design and implement a solution to the problem. You will gain marks for all of the following aspects. Note that you do not necessarily have to complete them in the order shown.



In this document, you will find:

- guidance on how much support and guidance your assessor can give you at each stage of the project
- information on what evidence you must produce for each aspect of the project
- guidance on choosing a suitable problem for your project
- guidance on how you might approach each aspect of the project

Support from your assessor

During your project, you are expected to work independently. However, this does not mean working entirely alone. Your assessor can provide you with guidance to help develop your thinking as you progress. This might be through general support in class on broad areas, such as research skills or project

planning, or individually in the form of constructive questioning and/or comments at face-to-face meetings or constructive comments on your work. This level of support is considered reasonable assistance within the marking guideline and will not affect your final mark.

However, your assessor cannot tell you specifically how to proceed with your project, how to re-phrase or improve responses or provide you with model answers. If you do require this level of support for an aspect of your project, this is considered to be beyond reasonable assistance. Where more than reasonable assistance is given, this will be reflected in the mark awarded for that aspect of the project.

Evidence to be gathered

You will have to ensure that you gather the following evidence as it is all required for this assessment:

- your detailed project plan, requirements specification and test plan
- your 'record of progress' through the project, including reflective commentary and all items of evidence specified within the task (including program listings, screenshots, web page source files, data files or similar as appropriate)
- evidence of your testing
- qualitative evaluation of your solution and development process

Before you begin

Before you begin, you might find it helpful to read SQA's publication 'Your Coursework' which your assessor should be able to give you.

You can download a copy from the Coursework page on SQA's website.

Guidance on recording your progress

You will probably be working on this project for several months, and during that time you will be producing evidence of many types. You need some method of keeping track of your progress and some way of organising all the evidence you produce.

One way to do this is to keep a regular diary of work you have done. This might consist of written notes kept in a notebook or loose-leaf folder, or an electronic diary or blog. It is a working record of your progress to help you to be organised and keep track of what you have achieved.

However you choose to keep this 'record of progress', it should include:

- brief regular notes on any work you have done (daily or weekly)
- notes describing any help you required
- **reflective and evaluative commentary on your progress**

Although it is not a formal report, your assessor will refer to your 'record of progress' when awarding marks so it is important that you regularly update it with relevant comments relating to the progress of your project and the decisions you make throughout.

It is useful to link your 'record of progress' to any evidence you produce, which needs to be stored methodically in a folder, ring-binder and/or electronic storage system. This could include annotated printouts, sketches, screenshots, program listings or similar, showing evidence of ongoing development and refinement of your planning, design and solution. These could be kept in order in the same loose-leaf folder, or if you are using an electronic blog, you could make links from it to the items of evidence. All evidence should be labelled, dated and annotated as it is produced.

You should keep this 'record of progress' up to date **throughout the whole project,** especially during initial research and analysis and during the design and implementation phases of work. It is important that you update it regularly so that you can keep track of your work over the duration of the project. You will definitely need to refer to reflective entries made in your 'record of progress' when evaluating your project, so it is vital to remember to record these at the time, as you may not be able to remember them later when writing your evaluation report.

Your notes and the reflective comments will contribute to the marks for some aspects of the project, so it is an important part of the evidence that your assessor needs to see. The marks awarded will be based on **all** of the evidence you produce during the project, including entries in your 'record of progress'.

Make sure you date and label all evidence, and store it in a safe place. The instructions and guidance which follow clearly indicate the evidence that you must retain for assessment.

You should discuss your progress with your assessor at regular intervals and at all main stages of the project.

Choosing a suitable problem

The project offers you an opportunity to develop your knowledge of Computing Science at Advanced Higher level and to apply this knowledge to a topic of interest to you.

The project requires you to work independently and to develop skills which will be invaluable to you in future study or the world of work. These skills include planning, research and analysis, problem solving, presentation and evaluation.

Choosing a project topic

You are expected to choose a suitable project topic. You may already have an idea or you may wish to explore ideas with your fellow candidates and/or assessor. You may also get ideas from online resources, industry news, television, local business partners, STEM ambassadors, etc. A successful project is likely to be in an area in which you have a genuine interest, either from a study, future career or personal point of view.

There is a list of possible project ideas at the end of this document. You could choose or adapt one of these, or use an idea of your own.

It may be possible to complete some projects within your centre, but you could also consider a project that might require collaboration with a university, college or local industry. Your assessor should be able to advise you about this.

Project criteria

The project must clearly demonstrate significant application of knowledge and/or skills at Advanced Higher level, related to software and information system design and development (as defined in the 'Further mandatory information on Course coverage' section of the *Course Assessment Specification*).

Your project solution must include the use of an appropriate range of techniques at Advanced Higher level Course, including:

- an interface appropriate for your users that validates all inputs

AND

- interfacing with stored data

AND

a minimum of **two** from the following:

- 2-D arrays, arrays of records or linked lists
- a binary search, a sort algorithm or other coding of similar complexity
- recursion
- HTML form processing using server-side scripting
- appropriate SQL operations

Project ideas checklist

When formulating your project ideas you should consider the following checklist of questions:

1) Will your project involve designing and developing a solution with an appropriate user interface that validates all inputs?

yes	no
☐	☐

2) Will your project involve designing and developing a solution that interfaces with stored data?

yes	no
☐	☐

3) Will your project idea involve implementing a minimum of two from the following list?

yes	no
☐	☐

- 2-D arrays, arrays of records or linked lists
- a binary search, a sort algorithm or coding of similar complexity
- recursion
- HTML form processing using server-side scripting
- appropriate SQL operations

4) Will your project idea allow you to apply generic skills in planning, research/analysis, problem solving and evaluation?

yes	no
☐	☐

5) Will you be able to complete it in the time available?

yes	no
☐	☐

6) Can all potential barriers to you carrying out your project be overcome, eg health and safety issues, permissions, logistics?

yes	no
☐	☐

7) Do you have access to any necessary expertise, resources and equipment (eg hardware, software, etc)?

yes	no
☐	☐

If you answer 'No' to any of the above questions, you will need to reconsider your project idea.

You may wish to keep an annotated copy of this page.

Producing a project proposal

Once you have decided on a project idea, you should produce a brief **project proposal.**

Your proposal should include a brief description of:

- your project idea
- the end-user group
- how it fulfils the requirements detailed in the previous section (requires relevant knowledge and skills, develops generic skills, feasible within timescale, all barriers to completion can be overcome and all required expertise, resources and equipment are available)

Your project proposal should be short, probably a couple of paragraphs, and certainly less than a page in length.

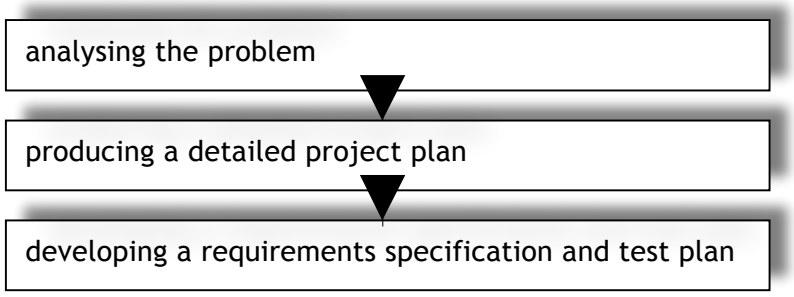
When you have written your proposal, you must discuss it with your assessor and obtain approval to continue.

Keep your project proposal as evidence, adding any reflective commentary that might be useful later.

Next steps

Once your assessor has approved your project proposal, you are ready to begin work on your project.

Before you begin the actual design and development of a solution, you must complete the following preliminary stages:



Analysing the problem

Before you start designing and developing a solution, you must analyse the problem that you are addressing, to ensure that you fully understand every aspect of it.

This analysis of the problem will require some research and preliminary work to find all the information you require. This should involve:

- a survey of end users
- an assessment of end-user requirements
- a brief feasibility study including
  - determining if there are any copyright issues
  - considering any software, hardware or other resources required
  - finding out about any additional techniques that you might require to implement your solution

Decide now what questions you need to answer, and what you need to do, then methodically carry out all the analysis and research required to find the information you need.

You should keep detailed notes on any research and analysis that you carry out, as you will use this information later to create a requirements specification.

You are also likely to need to carry out more research or analysis at later stages, when you encounter new problems.

Keep your notes of both your initial and any additional research in your 'record of progress'.

Guidance on research

Your research may include a variety of approaches, which fall into two main categories: primary research and secondary research. You will undertake at least one of these research methods during your project.

Primary research is information which you gather yourself by using one or more of the following: user surveys, interviews, site visits, company visits,
work experience/shadowing or other similar methods. When carrying out primary research it is important to:

- be polite and courteous at all times
- prepare appropriate questions in advance and have them checked by your assessor
- have a clear purpose for any visits or interviews
- discuss and agree in advance the terms of your visit with the people concerned
- be aware of any confidentiality or copyright issues, and obtain appropriate permissions
- comply with good practice in terms of research ethics (see additional guidance at the end of this document)

Secondary research is information which you gather from research already done by others and includes information from online sources, publications, trade magazines, specialist organisations, sector specialists, STEM ambassadors, learned bodies and different types of media. Critical evaluation of all information accessed would include checks on currency, authority, accuracy and balance.

You must record and reference any information gathered from sources like those described above. Your assessor or mentor can tell you how it's done, but here are a few pointers:

- Use "quotation marks" around any text that has come from other sources and identify clearly what those sources are.
- The sources of diagrams, illustrations and images need to be acknowledged.
- Write a bibliography (a list of references including all sources).
- Use a recognised referencing system, for example the Harvard referencing system.

Remember that plagiarism (passing off other people's work and ideas as your own) is cheating. This includes copying passages from the internet and not acknowledging the source.

It is also plagiarism if:

- all or some of your project has been produced by someone else
- it is example work that's been prepared by your assessor or mentor
- you have copied it from a book, an internet site or an essay bank without referencing it
- you have copied and used code from an unacknowledged source

Markers can easily spot plagiarism. It will lose you marks or may result in your qualifications being cancelled.

Producing a detailed project plan (10 marks)

Before you start any further work, you should produce a **project plan**. You can start with a fairly basic outline project plan.

This should identify and list the main stages of the project, and include an estimated time line for implementation. You will add more detail later.

Your outline project plan could be a simple table like the one shown below:

task	time	target date
requirements specification		
test plan		
interface design		
program/data structure design		
implementation		
final testing		
evaluation		
final submission of project		

Ask your assessor about the final submission date, and insert an appropriate target date for each stage which will enable you to complete your project in good time.

So far, your outline project plan only gives target dates for the main stages.

Next, you need to develop a **full and detailed project plan** covering resource requirements as well as time allocations. It should include all identifiable sub-tasks and intermediate targets, especially for the implementation aspect of your project.

You may find that you need to adjust your plan as your project develops, and you should keep a record of this with reasons for the changes. You will need to review your plan regularly with your assessor.

Assessment requirements

Your detailed project plan should include:

- **full details of all tasks and sub-tasks, resources required, time allocation and intermediate targets**
- **evidence of ongoing refinement of your plan**
- **detailed reflective commentary justifying changes to your plan**

Once you have completed your detailed project plan, you may want to update your notes or entries in your 'record of progress'.

From now on, you will use your **detailed project plan** to monitor and record progress. However, you should keep the **outline** project plan as evidence of your initial planning.

Guidance on project planning

There are several important aspects to a project plan and these include:

- dividing the overall task into manageable sub-tasks (note 1)
- estimating realistic times for carrying out these tasks (note 2)
- identifying tasks that you would have to complete before you could move on
- making allowances for contingencies and unexpected events and problems
- setting up a system for monitoring progress and managing your time (note 3)
- identifying and sourcing any specialised/specialist resources and equipment (note 4)
- addressing potential barriers, eg health and safety issues, permissions and logistics (note 5)

Note 1: sub-tasks
Sub-tasks within the design and implementation stages might include:

- further research
- refinement of coding
- ongoing testing

Note 2: realistic time estimates
You already know the final deadline for the project, as this was included in your outline time plan. You must now estimate realistic times for all the sub-tasks which you will have to complete. Remember to allow for holidays, or other events that will affect how much time you can spend on your project. For example, there may be weeks when you have assessment for other subjects, or are involved in some events. You should also factor in some time for contingencies (unexpected problems) which might occur and slow your progress.

Note 3: monitoring progress
To monitor progress and manage your time, you can use a Gantt chart or similar project management tool. There are free examples of these online. Whichever tool or method you use, it should allow you to set targets for completion of each sub-task and enable you to track progress through the project against these target dates.

Note 4: resource management
For this part of your project plan, you should create a list of all specialised resources you will need and when you will need them, for example: access to computer hardware, software or applications. Some of these may be available at any time, while others might only be available at certain times, so

you need to plan in advance to ensure that your project is not held up at any stage waiting for any resource.

Note 5: potential barriers
You will need to discuss your plans with your assessor, to identify if there may be barriers to doing anything that you have planned. For example, you may need to use specialised equipment which can only be used under supervision. If you have to go out of your centre to visit any workplaces, sites or other educational establishments, you need to check how you will get there, and whether you need to get permission. In all these things, it is important to have discussed them well in advance, so that you are not delayed at a later stage.

Producing a requirements specification (10 marks)

Once you have researched and analysed the problem, and created a project plan, you should put together a requirements specification for a solution.

The requirements specification outlines what the solution will do, describing what the functional and non-functional requirements are. It allows you to outline and assess all the requirements for your project development before beginning your design. It should provide enough detail to allow the design and development to go ahead. This should help to minimise later redesign (if any), and give you a realistic basis for scheduling your time and resources (project plan).

Assessment requirements

For this aspect of the project you should produce the following evidence:

- **a complete and detailed requirements specification, defining all user requirements and functionality, clearly based on findings from initial analysis and research**

Once you have completed your requirements specification, you should:

- **update your 'record of progress', including reflective commentary, as required**

Guidance on producing a requirements specification

A requirements specification is a document that defines all the important parameters of the solution. It is likely to include information about the following:

- high-level description of the purpose of the solution
- scope and boundaries of the solution
- clear description of the end users
- all user requirements
- all the functionality required by the solution (what it must be able to do)
- all inputs and outputs

Producing a test plan (10 marks)

Once you have put together a requirements specification for a solution, you should create a test plan.

Assessment requirements

For this aspect of the project you should produce the following evidence:

- **a comprehensive test plan identifying in detail all testing required**

Once you have completed your requirements specification and test plan, you should:

- **update your 'record of progress', including reflective commentary, as required**

Guidance on producing a test plan

Your test plan should be based on the requirements specification, and provide the basis for the final testing of the implemented solution. It should be systematic (following a logical plan) and comprehensive (testing all requirements).

It should include:

- what requirements will be tested/verified at each stage
- all the tests that will be carried out
- test data and criteria to be used
- how test results and conclusions will be recorded
- end-user testing

Some aspects of the requirements specification and test plan may need to be adapted as you work on your project. When this is required, you should keep all versions and make notes about the changes and the reasons for them in your 'record of progress'.

Iterative design and development of solution

By now, you should have:

- analysed the problem
- carried out any initial research that you need
- developed a detailed project plan
- produced a requirements specification and a test plan for your solution

Now it is time to start designing and implementing your solution.

There are three key aspects within the iterative development process:

- interface and program/data structure design
- interface and program/data structure implementation
- ongoing testing

These will all probably overlap; you don't necessarily have to do one after the other, although your design should inform your implementation and testing may require changes to the design, so the whole process is likely to be very iterative.

For clarity, however, this guidance describes them as three separate phases.

Interface design (10 marks)

The first aspect in the iterative development of your solution is designing the interface.

Assessment requirements

For this aspect of the project you should produce the following evidence:

- **a complete and detailed user-centred interface design, meeting the requirements specification**
- **use of a recognised design notation/methodology**
- **show validation criteria for all inputs**
- **detailed reflective commentary justifying all decisions made**

Throughout the design and implementation stages of your project, you should:

- **update your project plan regularly, including reflective commentary, to show progress towards project plan targets**

Guidance on interface design

Your interface design should be user-centred, reflecting your findings from your research and analysis including user requirements. It should also show what inputs and outputs are expected and how these will be implemented. You should complete your detailed design using a recognised design notation and/or methodology.

Evidence should include:

- annotated screen layouts
- an indication of how different screen layouts are interrelated
- a description of what validation is necessary
- a description of the purpose and function of all screen elements

Your reflective commentary should include:

- justification of initial decisions made with reference to:
  - end users
  - principles of good user-centred design
  - requirements specification
- justification of changes made as result of iterative development

Program/data structure design (10 marks)

The second aspect of the development of your solution is the design of any coding and data structure.

Assessment requirements

For this aspect of the project you should produce the following evidence:

- **complete and detailed program/data structure design meeting all requirements specification**
- **use of a recognised design notation/methodology**
- **detailed reflective commentary justifying all decisions made**
- **a record of any changes you make to your design as a result of implementation and testing**

Throughout the design and implementation stages of your project, you should:

- **update your project plan regularly to show progress towards project plan targets**

Guidance on program/data structure design

Complete and detailed program/data structure design should include:

- top-level design using pseudocode or other contemporary design notation
- refinement of sub-programs, as required
- an indication of the data structures (variables/files) to be used
- data flow between modules and/or sub-programs
- design of any coding for scripting, SQL or complex queries

Your reflective commentary should include:

- justification of initial decisions made with reference to:
  - end users
  - requirements specification
- justification of changes made as a result of iterative development

Implementation (20 marks)

You are now ready to complete the implementation of the interface and program/data structure as you build your solution.

Assessment requirements

For this aspect of the project you should produce the following evidence:

- **evidence of complete implementation of interface and program/data structure design to meet all requirements specification**
- **use of a range of techniques appropriate to Advanced Higher level**
- **evidence of ongoing testing, correcting and/or refinement (including annotated listings and screenshots)**
- **detailed reflective commentary justifying all decisions made**

Throughout the design and implementation stages of your project, you should:

- **update your project plan regularly to show progress towards project plan targets**

Guidance on implementation

Implementation should be based on your designs, and so meet all aspects of the requirements specification.

You need to ensure that you are using an appropriate range of techniques at Advanced Higher level, which should include at least two from:

- 2-D arrays ,arrays of records or linked lists
- a binary search, a sort algorithm or other coding of similar complexity
- recursion
- HTML form processing using server-side scripting
- appropriate SQL operations

Throughout the implementation process you should test the component parts as you go and maintain evidence of ongoing tests, noting any corrections and/or refinement you make as a result. Your notes should reflect and justify the decisions you have made.

Final testing of the solution (10 marks)

If you have completed the implementation of all aspects of your solution, then you are ready to carry out final testing of the solution. This testing should be systematic and comprehensive, and based on the test plan you produced earlier.

Assessment requirements

For this aspect of the project you should produce the following evidence:

- **evidence of comprehensive testing of completed solution against all elements of test plan, with results showing that the solution meets all requirements specification**
- **clear evidence of systematic user testing**
- **clear evidence of any final corrective maintenance (if required)**
- **detailed entries in 'record of progress', including reflective commentary**

As you undertake final testing and carry out any required corrective maintenance, you should:

- **update your project plan to show progress towards project plan targets**

Guidance on testing

Final testing should be based on the test plan you created early in the project.

Make sure you carry out all the required tests, and record the results in a structured way.

If your solution fails any of the tests, you will need to consider what action to take. This will depend on the time you have available, and the nature of the problem. These may be simple bugs that can be fixed quite easily, or they may indicate deeper design flaws.

If you have enough time, you should carry out any corrective maintenance or improvements required, and make some notes on what you have done. If you discover serious problems, you may need to return to some aspects of the design first, before implementing changes.

If you identify problems that you do not have time to correct, don't worry — you should make detailed notes about the nature of the problem, with an indication of what needs to be done to solve it.

Evaluating the solution and the development process (10 marks)

You have now completed the iterative design and development phase, tested your solution and implemented any corrections required, but your work is not yet done. You must now evaluate what you have done.

Assessment requirements

You should produce the following evidence:

- **complete and detailed qualitative evaluation of the solution, referring to requirements specification, results of final testing and user testing**
- **relevant and justified evaluative commentary on the development process and your own performance, referring to reflective comments in your 'record of progress'**
- **detailed description of possible further developments**

Although you have now completed the implementation phase of your project, you should continue to:

- **update your project plan to show progress towards project plan targets**
- **update your 'record of progress', including reflective commentary**

Guidance on evaluating

Evaluating the solution
When evaluating the solution, you should report on qualitative and performance aspects.

Your evaluation could include:

- commentary on how closely the solution matches the requirements specification
- results of final testing and user testing
- commentary on how well it matches end-user requirements

You should include some valid and relevant conclusions based on your evaluation.

Evaluating the development process and your own performance

You are also required to evaluate:

- the development process that you have followed in undertaking your project
- your own performance

As well as descriptive elements, your evaluation of the process and your own performance should also reflect on:

- what went well
- what went not so well
- what you might have done differently
- what you have learned
- your responses to feedback

You should find that the reflective comments you have recorded in your 'record of progress' will help you to evaluate the process and your own performance.

Describing further possible developments
You should reflect on the work you have done and describe in detail some possible further developments of your solution which could be undertaken.

Note that you will **not** be required to implement any of these!

Final check

You have now come to the end of your project.

Check to ensure you have all the required evidence, properly organised.

If you are sure that you have finished, let you assessor know.

Guidance on research ethics

All universities issue ethical guidelines in order to protect the public, the researchers and the institution itself, and they require all research proposals to be submitted to an ethics committee for approval.

Universities publish on their websites guidelines on research ethics, which may be divided into research involving human subjects, research involving animals and research involving the environment. It may be useful to have a look at an example of these guidelines.

Companies and organisations also work within ethical guidelines when carrying out research. The following extract from '*Research Methods in Education*' provides some basic issues to consider with regard to research involving human subjects:

1. It is important for the researcher to reveal fully his or her identity and background.
2. The purpose and procedures of the research should be fully explained to the subjects at the outset.
3. The research and its ethical consequences should be seen from the subjects' and institution's point of view.
4. Ascertain whether the research benefits the subjects in any way.
5. Where necessary, ensure the research does not harm the subjects in any way.
6. Possible controversial findings need to be anticipated and when they ensue, handled with great sensitivity.
7. The research should be as objective as possible. This will require careful thought being given to the design, conduct and reporting of research.
8. Informed consent should be sought from all participants. All agreements reached at this stage should be honoured.
9. Sometimes it is desirable to obtain informed consent in writing.
10. Subjects should have the option to refuse to take part and know this; and the right to terminate their involvement and know this also.
11. Arrangements should be made during initial contacts to provide feedback for those requesting it. It may take the form of a written summary of findings.
12. The dignity, privacy and interests of the participants should be respected. Subsequent privacy of the subjects after the research is completed should be guaranteed (non-traceability).
13. When ethical dilemmas arise, the researcher may need to consult other researchers or teachers.

Adapted from Reynolds, P.D. (1979) 'Ethical Dilemmas and Social Science Research' in L. Cohen, L. Manion, K. Morrision, *Research Methods in Education (5th Edition)*, (2005, San Francisco: Routledge Falmer)

Additional guidance: project ideas

Here are some examples of possible project ideas of suitable complexity that you might like to consider:

Project idea	Examples of potential users
Design and develop a program/information system that processes, stores and allows updating of sporting or other results.	a school sports day for school sports co-ordinators, PE department, local sports council, local club
Design and develop a 2-dimensional game with ability to pause and store state of the game and restart, such as Connect4, Battleships, Minesweeper, Chess, Draughts, Sudoku, etc.	computing science students, primary groups, teaching staff
Design and develop a small social media website that allows registered users to blog and share information.	special interest groups, such as fashion followers, gamers, fan clubs
Design and develop a database-driven website that allows users to search content and display, edit and update it appropriately.	customers booking holiday rentals
Design and develop a multi-player card game, such as 'Top Trumps', or a card-matching game.	primary school pupils studying a specific topic, or nursery school pupils matching colours or objects
Design and develop a question-based quiz that randomises questions, processes and stores results, for example 'Who wants to be a Millionaire', 'Mastermind', 'Pointless'.	computing science students, primary groups, teaching staff
Design and develop a simulation of the operation of a queue, a stack or a sort algorithm, or other similar computing science concept/process.	computing science students, teaching staff
Design and develop an interactive computer assisted learning tool that gathers and stores information.	language, science, mathematics, computing science students at primary or secondary schools, or similar