The project reports should be coherent documents giving an integrated account of how you solved the given problem, the results you found, and the overall conclusions that can be drawn from them.

- The intended audience that you should aim for in your writing style is someone who is assumed to know enough about the problem to have asked it in the first place, and to be reasonably technically literate without necessarily having the exact expertise or time to solve the problem themselves. You can also assume the reader has read the project description, so you should not need to repeat this information in detail.

You should ask yourself how you can express your methods and conclusions as clearly as possible, and try to imagine what kind of information such a recipient of your report is likely to need.

- The lectures will provide introductory material on the techniques to be used and you may justifiably feel that some of this should be included in the write-up. You may need to source additional material from journal articles and library books. Please try to make an intelligent selection of what fits in with your narrative, rather than writing everything down.

- A typical structure for your report is indicated below. Output data and plots from the computer should be used to underpin and justify the main framework of the text – it is not adequate just to present numerical results and plots without any explanation or interpretation.

- A good length for a report is of the order of 8-15 pages, depending on how much computer output you want to include. This is only a guide, and the exact amount will depend both on the project itself and on your own personal style.

As a general framework for writing the reports, something like the following is suggested but is not required:

1. **Background.** Summarises relevant theory without repeating lectures.

2. **Description of program development.** Describe how you went about writing your program, and what particular problems you had to overcome on the way. The extent of this section will be very dependent on the type of project and what proportion of the program has been supplied. This may also be the appropriate place to describe any tests you performed.

3. **Answers to individual topics specified in the project.** This is where you should detail your responses to the individual sections of the project. Answer the questions as best you can, as coherently and clearly as possible (remember your intended audience). Include graphical or other output at this stage in such a way as to underline your argument or supply what is requested in the project sheet.

4. **Conclusion.** It is very important to summarise in perhaps half a page what you have learned overall by doing the project, and to discuss the main scientific and practical results from your investigation.
Grade Descriptors and Criteria (Faculty Scale)

Project work is notoriously difficult to evaluate. To give you any idea of the quality required for each level, the statements below are a condensation of slightly edited comments made on actual student reports by previous lecturers. Obviously students may have achieved higher levels on some aspects of the project and lower levels on others.

**Third Class 65–69** Too short, nearly all the specific questions from project sheet have been left unanswered or undiscussed. Results convey little information and there is no interpretation of their significance. Evidence that the project was rushed or started at the last minute. Serious errors in the numerical code, that could have been discovered by discussion with lecturer or peers. The graphs are plotted in a way which conceals errors in the solutions.

**Second Class (2nd Division) 70–74** No discussion of the most interesting aspects of project. Discussion of journal articles not coherent. Several parts have not been attempted, both analytical and numerical. Leaves out a lot more detail than it should.

**Second Class (1st Division) 75–79** Reasonable effort, with programming correct but discussion poor. Greater effort needed to explain your findings, and prove you understand what is going on. Computations presented with no explanation of the method. Many pictures, but no real message about what is going on. Impression that time ran out before the job properly finished. Insufficiently detailed, makes no effort to explore any further implications.

**First Class 80–89** Addresses all questions in report and proven decent programming skills. Topics treated fully and answers are correct and accurate. Good introduction that combines information from the lectures and also from other Honours and 3rd year courses. Poor choice of parameters prevented you from noticing a nice feature. No explanation of why your proposed method actually works, or a discussion why it’s preferable to an existing one. Failure to notice a known exact case which could have been used to evaluate results. Report lacks penetrative insight, description of journal article confusing and hard to follow. Programming style not elegant, but is effective. More insight could have been supplied into why the various things occurred and were interesting.

**First Class (with possible medal) 90–94** Concise writing style, but some bits ambiguous and reader has to make assumptions. Showed you understood the mathematics, could implement ideas in working code and explored new ideas. Not afraid to confront the problems to arise, though does not always solve them. Would be improved by a greater readiness to question assumptions and develop the material beyond what is literally given on the project sheet. Thorough and honest in describing author’s thought processes and misgivings.

**First Class (with medal) 95–100** Good programming skills, independent work and thinking and a creative approach to problem solving. Only errors are minor typographical, isolated and non-severe analytical errors or difficulty with very advanced concepts. Originality of approach or interpretation and a willingness to experiment. Trying a number of independently developed ideas even if not successful. Goes the extra mile, in use of additional algorithms, and attempt to understand why a method fails. Includes interesting additional feature. Achieves considerable gain in efficiency. Raises interesting and plausible questions for further investigation.