The aims of the report are (i) to present the work to a wider audience unfamiliar with the project, (ii) to provide a written record for future reference by users of your software and further developers of your work, and (iii) to demonstrate your skills in writing and structuring a report.

The report should give a clear description of the development approach use and must describe the use of appropriate tools to support the development process. It should give a full and accurate description of the work done and the achievements of the project, together with complete software documentation and a user manual and/or technical manual as appropriate. Every effort should be made to provide a professional, quality description of the work. After the examiners' meetings in May/June one copy of the report will be returned to the student.

Project reports from [past years](http://dacwww.computing.dundee.ac.uk/teaching/honours_projects/login.asp) are available.

### **Format of report**

The report should be formatted as a justified, double column, single-spaced, 10pt Times New Roman font document using an appropriate word processing system such as Microsoft Word, OpenOffice Writer or LaTeX and converted to a PDF file. The report should not exceed 15,000 words or 15-20 pages in length (excluding appendices). The report should be printed single-sided.

### **Binding of report**

The two hard copies of the report should be securely bound. One way to do this is to use the spiral binding machine in the Administration Office (QMB 1.03). There is a charge of £1.50 for this to cover the costs. If you are unable to get the binding done by 12.00 because of a queue for the binder, then you can just hand in two unbound copies and the admin staff will bind them for you. However it is helpful if as many students as possible do the binding themselves.

### **Example report**

There is **an example document** ([PDF version](http://dacwww.computing.dundee.ac.uk/courses/ac40001/ProjectFormatExample.pdf), [Word document version](http://dacwww.computing.dundee.ac.uk/courses/ac40001/ProjectFormatExample.doc)) which describes and illustrates the format which must be adopted for the report; please consult this for more detail. For those using Agile methods this **alternative version** will be more appropriate: ([PDF](http://dacwww.computing.dundee.ac.uk/courses/ac40001/AgileProjectFormatExample.pdf) , [Word](http://dacwww.computing.dundee.ac.uk/courses/ac40001/AgileProjectFormatExample.doc)).

### **Report style and content**

The report should be written in a formal style: it is neither a diary nor a magazine article. All pages should be numbered. **All references should be cited in the main body of the report**. The report should demonstrate that the student has used appropriate tools to support the development process and that **verification and validation have been applied at all stages**. The number of sections and their headings will vary from project to project. However, the main body of the report should address the following:

**Introduction:** An explanation of the problem and the objectives of the project. It is very important to give a clear description of what the project is actually intended to do, preferably in non-technical terms.

**Background:** A review of relevant literature and any similar products. The project should be placed in a wider context and this could include the scientific, technical, commercial, social and ethical context.

**Specification:** A specification of the problem and an explanation of how the student arrived at this specification. An initial work schedule including an overall project plan with time-scales, deliverables and resources. If using agile development, a prioritised product backlog.

**Design:** This should include the design method, design process and outcome. Design decisions and trade-offs should be described e.g. when selecting algorithms, data structures and implementation environments or when designing for usability.

**Implementation and Testing:** A description of production, testing and debugging. A demonstration (or even a proof) that the specification has been satisfied.

**Evaluation:** Usability should be evaluated with a description of the user-centred design methods employed to produce a usable product, including rapid prototyping, usability methods, results and re-designs as appropriate. Other relevant criteria such as accuracy and computational efficiency should also be employed for evaluation as appropriate.

**[Sprints:** If you are employing agile methods, the three topics above (Design, Implementation and Testing, Evaluation) should be incorporated into each sprint, with evaluation normally being the sprint review.]

**Description of the final product**: A clear description of what the final product looks like and what it does. **This is vital but often neglected.**

**Appraisal:** A critical appraisal of the project indicating the rationale for design/implementation decisions, lessons learnt during the course of the project and an evaluation (with hindsight) of the final product and the process of its production (including a review of the plan and any deviations from it).

**A description of any research hypothesis**

**Summary and Conclusions**

**Recommendations for future work**

A copy of the mid-project progress report should be included.

(Derived from: *British Computer Society, Guidelines on Course Accreditation, 2006, Section 2.2.6: Individual projects*; *British Computer Society, Guidelines on Exemption and Accreditation, August 2001, Section 2.4: Project Work*).

## **Appendices**

The main body of the report should read as a self-contained document. However, appendices can be used for necessary supporting documentation. You should submit software, source code, a user manual and/or technical manual as appropriate and minutes of your meetings as appendices on a disc. These can be in separate files arrange appropriately in different folders; there is no need to compile them into a single document. Portable Document Format (pdf) should be used for all documents such as user manuals. Hardcopies of minutes and source code are not required.