UG Project Manual

Overview

The Purpose of the Project

Your Honours project has several objectives:

- to give you the opportunity to undertake a substantial programming project, and to draw together the many skills you have acquired in the previous three years;
- to give you practice in writing good technical English by writing a report of the project undertaken;
- to give you instruction in a field not, or perhaps only lightly, covered elsewhere in your courses;
- to act as one part of the assessment of your degree.

The Role of the Supervisor

Typically, you will be working on a topic suggested by a supervisor. The supervisor will have certain expectations of your project, and will be on hand throughout the year to discuss progress and to guide you through the project. However, you should remember that this will be your Honours project, and the best projects are normally those that are driven by the students and which contain ideas that they have contributed. The precise methods of supervising vary from one staff member to another, but normally you will have an arranged hourly meeting once a week with your supervisor. It is important that you keep your supervisor up-to-date on the progress you are making. If you encounter any difficulties or other problems which are hindering your progress, you should inform your supervisor in the first instance. If these problems cannot be resolved directly, you should discuss them with the Level Four Teaching Coordinator.

What's Involved?

The first stage of the project will likely involve mastering any new skills needed. For example, you may need to learn a new language or package, or learn about some piece of hardware. As part of this first stage you will be required to submit a Project Plan. The plan may be adjusted later as the project unfolds. The bulk of your time will then be spent developing your ideas, implementing them and testing them. At the end of the project, you must submit the software you developed and a written report. It is important that you are able to communicate your ideas and explain your work. Hence, we strongly recommend that you keep a notebook or web log to record your weekly progress. As your project proceeds, you should also document your programs, including test programs and methods that you reject, because these will all be useful when you come to write your report. The marks for your software and report normally carry equal weights. Finally, you must submit an electronic archive of your software, and give a demonstration of your project to your markers.

Special Options and Joint Honours

Students intending to graduate with one of the special degree designations (such as Computing Science (Artificial Intelligence) or Computing Science (Business), for example) must take a project with sufficient content in the relevant area.

Where a project is considered suitable for one of the special options, it will be indicated beside the title in the list of staff suggestions. Students doing the standard Computing Science degree are eligible to do any listed project. If you are in doubt over the suitability of a project for your intended Honours degree, please ask.

Joint Honours students must also undertake a Computing Science project. However, this tends to be worth a total of credits different to the Single Honours project, and hence a Joint Honours project is not expected to be as substantial as a Single Honours project. As a guide, Single Honours students would be expected to spend around 30+ hours per week on their project, whereas Joint Honours students should spend around 20+ hours per week. Students should ensure they are clear what the credit weighting of their project is by visiting the course catalogue.

Please note that the timetable, deadlines, project allocation procedure, and submission requirements are normally exactly the same for all types of project. Supervisors are aware that Joint Honours students spend less time on their projects than Single Honours students and this is taken into account during marking.

Proposing Your Own Project

Most of the academic staff in computing will want to supervise projects working in their own area of interest. This makes sense because this is where they have invested their time to gather expertise. Also, they will want to have a team of people working on the same area because this is the best way to make progress in a research area. For these reasons many staff will not be keen on supervising projects outside their specialist areas. However, if you have ideas for your own project, it is possible that you might be able find an appropriate member of staff who would be prepared to supervise your project. To help the member of staff to assess whether your idea is suitable, you should write a short project proposal (2 pages max) to outline the idea, describe the hardware and software resources required, and to highlight any potential pitfalls. If the member of staff would be willing to supervise your project, you should then ask that member of staff to send your project proposal to the Level 4 UG Project Coordinator and to tell the coordinator they are happy to supervise the project. This way we can have a very clear idea and record of the agreement to supervise the project.

There is no guarantee you will get allocated to the supervisor you want though - so if you want to increase the likelihood that you get to do your project then you will need to discuss your project with a supervisor as soon as possible (ideally within the first few weeks of your first term).

Please note, projects that rely on special hardware or software which is not readily available or installable in the department generally cannot be supported. Similarly, projects that would require you having elevated privileges on university computers cannot be supported (for obvious reasons!). However, it is possible to arrange for some dedicated project machines to be set up and allocated to specific students if they require specific drivers or other available hardware to be installed. If in doubt, please consult your potential supervisor or the Level 4 Coordinator.

Note that we've had some pretty crazy proposals in 2010, some involving using robots to do various things. We don't have any special robots, except for those you've seen in undergrad courses, and it is not possible for us to buy expensive robots for someone's project idea.

The Project Plan

Once you have started your project, the first deliverable is the Project Plan. Your Project Plan, which must be drawn up in consultation with your supervisor, should set out what you intend to do and when you intend to do it. Typically, a Project Plan would describe a number of achievable goals, along with a few further "nice-to-have" features or options. Initially, the plan will also be used by the Department to ensure that you are starting your project in a sensible manner, and that you are proposing to tackle a project that has sufficient content for an honours degree, but which is not too ambitious. Subsequently, you can use your plan as a yardstick to check your progress.

Your plan should be no more than *four* typed A4 pages and should contain:

- a brief introduction to the project, giving the background and explaining why it is a worthwhile task;
- the main goals, making it clear which goals are central, and which are optional extras;
- risk assessment, describing likely circumstances in which the project would become unfeasible, and the steps that would be required to recover the project;
- a list of the hardware, software, and any other resources required;
- an outline timetable.

NEW: You also need to check out if your project needs an ethics review, and fill the required forms if appropriate. Check here

Ideally, a plan will include a basic project, which should be fairly straightforward to implement, and which should be completed reasonably early. It should also include more ambitious extensions, or possible developments to be tackled depending on the results of your initial work. A well planned project should not be one which only comes together at the end of the period.

An example template project plan document, and a tar file of the corresponding LaTeX source (on Unix, just type "make") is available within the Course Materials session of MyAberdeen.

Submitting Your Project Plan

You should submit your Project plan before the deadline. Once you have agreed your plan with your supervisor, please then:

- submit it on MyAberdeen, as either a Word or PDF attachment.

Assessment

The UG final year project is composed of the following assessment elements making up the total mark.

- Report/Dissertation
- Poster

Note. Weightings for these elements are provided in the assessment section of the MyAberdeen. Marking of the report will also consider your technical presentation/demo and code listing.

The Technical Presentation/Demo

You will be required to give an oral presentation of your project, and demonstration of your software. **This is not a formal presentation in front of a large audience** , it's a simple explanation of your project, where you may sit with your computer and present just to the *two members of staff who are marking your project.* You should use your own computer to demonstrate your software via video conferencing. ~~You may use your own computer to demonstrate your software, or any of the machines in the University labs.~~ There are two main reasons for the demonstration: to give the markers a feel for what the software does and how it looks, and to demonstrate that your software does what you claim in the report.

The time and date can be arranged between yourself and your markers, any time the week before the deadline of the report. You should e-mail your markers to arrange this. The demo/presentation should last *at most* 30 minutes.

The presentation/demo is taken into consideration as part of the total marks for your dissertation. You will be marked on how well you present your project, and not on the content of the project itself (although we will take into account subjects which are hard to explain). In your presentation, you will be expected to:

- give an introduction to your project;
- say why it has been interesting or worth doing;
- describe your main achievements, and any problems you had (and how you solved them!).

In the demonstration, you should give a brief overview of your software, and then take the markers through its use step-by-step. Remember that you will know the software and the interface inside out, but the markers may never have seen it before. So, don't dive in to the details before you have motivated it, and don't charge through a sequence of screens and dialogue boxes without letting the markers see what you are typing and explaining why you are typing it. You should be prepared to answer a few questions about the software, and about the decisions you took, and be prepared for the markers to test aspects of the software for themselves.

Preparing your Presentation

Aim for about 15 minutes of presentation, you will need to think clearly about what you can get over to your markers in such a short time. Do not try to put in too much detail. Make good use of slides and diagrams. Even though one diagram may be worth a thousand words, don't try to get too much on one slide. The marker should get a clear idea of what it was about, what issues you were tackling, how you set about it, and what degree of success you had. In the past we have observed that too many people are so involved with the minutiae of their project work that they give presentations which leave the markers with no clear picture of what is going on. Use the preparation of your presentation as a chance to stand back from the actual day to day work you are doing, and think about the issues you are facing.

The links below contain some advice and suggestions for giving presentations:

- Dealing with Nerves
- Planning your Presentation
- Using Visual Aids
- Assessing your Audience
- Self-Assessment

When we mark the presentation/demo, we will be looking at:

- Did they introduce and motivate the project, so that a non-specialist could understand the goals?
- Did they speak clearly, making eye contact, and hold our attention during the talk?
- Were the slides well-prepared and helpful?
- Did they manage their time well?
- Did they handle the questions well?

The Poster Presentation/Demo

Along with a dissertation (see below), you are expected to create a poster of your project . Guidelines about the development of posters are available in MyAberdeen.

The Dissertation

During your project, you might spend the bulk of your time on implementation. However, it is very important that you are able to communicate your ideas and to make your work understandable to others. At end of your project, you may know what you have done but the examiners can only judge the quality of your work by the description you give in your report. Hence you should make sure your written work is presented to a high standard and that it fully describes all the work you have done.

The Project Report

Your Report is primarily a description of what your project was all about. However, it should not be a blow-by-blow account of everything you did or tried to do, nor should it describe every line or subroutine in your code. By now, you should be able to write easily understandable programs. The explanations of the workings of your program(s) should be restricted to an overall description. You should aim to keep details down to an essential minimum, although you should highlight any novel, difficult, or unusual aspects of your project or code. Remember that other people will be reading your report - people who know nothing about the project, and even people who know little about the topic. Therefore, the description should be clear to a non-expert reader, and you should do your best to make it readable, understandable, and interesting.

Declaration

Put this declaration at the beginning of your report: *I declare that this document and the accompanying code has been composed by myself, and describes my own* *work, unless otherwise acknowledged in the text. It has not been accepted in any previous application for a degree. All verbatim extracts have been distinguished by quotation marks, and all sources of information have been specifically acknowledged.*

Word Limit

Your report should be 15,000 to 17,000 words in length, which is roughly 50 pages at 340 words per page (not counting diagrams, etc.). You may lose marks for exceeding this limit. The idea of this is to encourage you to be concise in technical writing and to get all the important information across and skip anything not relevant. The word limit just applies to the main body of the report, and does not count table of contents or references or appendices etc. You can put whatever you like in appendices, but markers are not obliged to read them.

The text of your report should be in a 12 point font with a spacing factor of 1.5 times normal. Common choices of font are Times New Roman or Arial, for example. Your report should include an Abstract, appropriate Acknowledgements, a Bibliography, and a signed declaration stating that the work is your own. A typical declaration is given below:

We do not specify a precise structure for your report. This will depend on the nature of your project. You should discuss possible report structures with your supervisor, who should be able to show you examples of previous reports. In general, you should divide your report into chapters, and these should include at least an Introduction and a Conclusion chapter. The Introduction chapter should describe the background and motivation for your project, and it would typically include a review of the appropriate literature, and your aims and objectives. It could also mention design decisions, and implementation issues, although the details of these would normally go into a separate chapter or chapters. The final Conclusion chapter should contain a critical discussion of your project. This should include a discussion of what went well and what went not so well about your project - this could range from discussing good and bad design decisions to the quality of your supervisor's advice! You could also discuss what you might do differently next time, and you should certainly give some suggestions for future work.

Using a Bibliography and References

At the end of your report, you should include a bibliography which lists the articles and books that you have cited in the main text. Each entry should allow the reader to identify and obtain the source. Therefore, you should include as much as possible of the following information: authors' names, article or book title, journal or collection title, editors, publisher, place of publication, date of publication, and page numbers. If you are referring to a web page, you should also include its full URL.

In the main text, you should be consistent in the manner in which you cite articles in the bibliography. There are many possible formatting conventions for citations and bibliographies. For example, the entries in the bibliography may be ordered alphabetically by first author name, or in order of citation from the text. They would then be cited from the main text by author name and publication year (e.g. A. Author et al., 1992), or numerically (e.g. [23]), as appropriate. However, once you have chosen a particular style, you should use it consistently throughout your report.

Using Figures and Tables

Using good figures and tables, including screen-shots, can greatly enhance the quality of your report. It is worth remembering that the Departmental PC classrooms have software which may well help you produce good figures. Generally, every figure and table in your report should have a unique number with which it is referenced from the main text, and it should have a short explanatory caption (even if it is also described in the main text) so that it can be understood in isolation. It is conventional to locate figures and tables between paragraphs soon after the point from which they are first referenced from the main text. Often, word-processing packages allow you to produce an index for the lists of Figures and Tables, in addition to the main Chapter/Section index.

The User Manual

A User Manual is required so that someone other than yourself can sit down at a machine and run your program and test its performance. It is a good idea to ask a friend to test the manual by trying to run the system from its instructions. The User Manual is often a good place to put screen-shots that show your program in action. **The User Manual must be added as an Appendix to the main dissertation and will therefore be marked as part of the dissertation.**

The Maintenance Manual

This should be used to describe the details of your implementation. It should be usable by people wanting to install the program, modify the program, extend the program, or trace bugs in its execution. This is an important part of the documentation, and you should ensure that you include details such as:

- instructions on how to install the system
- instructions on how to compile/build the system
- hardware/software dependencies, including libraries and other packages

- Organisation of system files, including directory structures, location of files within directories, details of any temporary files
- space and memory requirements
- list of source code files, with a summary of their role
- crucial constants, and their location in the code
- the main classes, procedures, methods or data structures
- file pathnames, particularly for accessing files of data values
- directions for future improvements
- bug reports

Again, the Maintenance Manual must be included as an appendix to the main report and will therefore be marked as part of the dissertation. (this is in addition to a copy of the maintenance manual with the code tar file)

Here is an example template project report document, and here is a tarfile of the corresponding LaTeX source (on Unix, just type “make”).

The PDF Code Listing

The Code Listing will be marked as part of the dissertation and should include a listing of all of your source code together with any necessary input data and output results. Remember that the information must be easy to read, so you should use an appropriate formatting program such as print or pr.

Submission Format

Every page of every document must be numbered so that everything in your report can be found easily. Near the beginning of your report there should be a table of contents page.

Last Week Rush

If you find yourself rushing to get things done for the last week your strategy should be to focus on getting a complete report with all major sections done. Set yourself a date like Wednesday for having a complete report, with everything important there - even if in a shoddy state. Then you have a couple of days for polishing. Even today you should be able to put together a complete report, with all planned sections/subsections, even if there are a few blanks here and there. Most people will find the deadline cuts short the work they planned to do. You just need to make sure that when the deadline comes you have something complete - even if not as perfect as you’d like. A bad mistake is to do some sections to a high standard and end up having to rush other sections at the last minute. It’s like being in an exam and spending too much time on doing an amazing question 1 and not having time for the rest. You need to cut short work on some sections, so you spend proportionate time on others.

Resources

There are a wealth of resources on the Internet dealing with how to write. A selection of these is provided below.

- Better Academic Research Writing, A Practical Guide
- Writing a (Computer Science) Paper

Submission Instructions

Your final project submission should consist of the following deliverables:

- Report PDF File
- Software Tar File
- Software Code Listing (PDF file)
- Conference Poster (PDF file)

Instructions on preparing and submitting these items are given in the sections below.

Making a PDF File of Your Report and Code

Create a PDF version of your report file called `surname_forename.pdf`, where "surname" is replaced by your last name and "forename" is replaced by your first name. If you wrote your report using Word, you can "save as" PDF.

You must also include a separate PDF file with all your code listing put together and properly presented, that is, to make it easy for the marker to browse through your code.

Preparing The Software Tar or zip File

You are asked to arrange your project files and folders under a single top-level folder called surname_forename, where, as before, "surname" is replaced by your last name and "forename" is replaced by your first name. You can just make a .zip file of it. or On a Unix machine, make a tar file of your software directory and compress it using `tar zcvf surname_forename.tgz surname_forename`.

This should give you a gzipped tar file called `surname_forename.tgz`, which is the file you will submit.

In addition to your project files, please ensure that the tar or zip file also contains a PDF version of the pages of your Maintenance Manual, and a simple text file called `readme.txt` that summarises the contents of your tar file. The `readme.txt` file should also mention any hardware or software dependencies (e.g., it requires Java and a particular IDE to be installed), and it should give sufficient instructions on how to compile and run your program.

In summary, your code tar file must contain at least the following:

- an explanatory `readme.txt` file
- your Maintenance Manual
- all of your source code, and any necessary data files
- a working executable of your program

It is generally not necessary to include other code that you may have used but did not modify yourself (e.g. code that could be downloaded from an external web site). However, you should mention the dependency on such external code in the list of hardware and software dependencies. If you have any doubts or questions about what you should include in your code tar file, please ask your supervisor. You shouldn't need to have a huge submission (gigabytes or more). If you have maybe you are including things you don't need to.

It is suggested that, prior to submission, you verify the contents of your tar file by unpacking it somewhere else, such as in a new directory under/tmp, for example:

```
bash$ mkdir /tmp/surname_forename
bash$ cp surname_forename.tgz /tmp/surname_forename
bash$ cd /tmp/surname_forename
bash$ tar zxvf surname_forename.tgz
bash$ ls -l surname_forename/\* Maintenance_Manual.pdf
... # your other files and folders shown here...
```

If possible, you should verify that your installation instructions make sense and that your program will still run properly if it is installed in an arbitrary folder (i.e. your code has no hard-wired absolute pathnames).

For your project demonstration, you should be prepared to demonstrate your software starting from your submitted tar file, if asked to do so.

Submitting the Final Deliverables

To submit your project files electronically use MyAberdeen.

Late Submission

Managing deadlines is an important part of the software development process. There are a number of deadlines associated with your project.

The first deadline for assessed work is the presentation date. You must give your presentation at the time specified - failing to do so without documented medical or compassionate reasons will result in a mark of zero for that component.

The deadline for the project submission is a “hard” one for which there are **no late submissions**! If you fail to submit your project by the set deadline without documented medical or compassionate reasons you will not have a chance to submit it later. You are reminded that that the project marks form a substantial component for deciding your overall degree classification.

Marking the Project

The project is marked by two members of staff, neither of whom were involved in the supervision of your project. The two markers will first mark your report individually and independently, and they will submit their marks to the departmental secretary. Once both marks have been received, the markers will then be asked to meet to agree on a final mark. If there is a significant difference between the original marks (i.e. they span a critical boundary (1st, 2.1, 2.2, 3rd, marginal fail, etc.), then the procedure is as follows: first the two markers each inspect the other’s report, and look at the categories where the marks are significantly different, and read the text comments for those categories. Then they discuss why they feel their comments are justified or they change their mind to take on board something they had not considered before. If the markers can agree on a change of position (by one or both markers) then there will be no need to call a third person. In the case of agreement, a summary of the outcome of the discussion will be written on the front of the first marker’s form. If they cannot agree on a mark, a moderator (that is, a third member of staff) will be called in to determine a final mark, which must lie between the original two marks. The external examiner will also consider the project, and the final mark will be agreed and approved by the Examination Board.

Marking Scheme

Firstly, the work undertaken in your project must be substantial. That is, the amount of work should be compatible with the number of credits (and nominal effort hours) of the course. We encourage students to look at past reports to get a “feel” for the quality and amount of work involved in First Class projects.

Additionally, following the BCS Accreditation Requirements (section 2.2.7) the project must provide evidence of your ability to apply practical and analytical skills. Some other project requirements of the BCS code of practice are:

- Investigation of the subject area
- Clearly defined research methods
- Specification of project aims and outcomes

- Development of a solution to a practical problem which involves the development of new software
- Follow a structured design process, involving s number of design stages
- Use of appropriate tools to support the development process
- Description of the verification and validation used
- Consider quality, reliability, maintainability, etc, as appropriate
- Consider the research and design methodologies, as well as the product or deliverable
- Include a critical appraisal of the project

The markers' report form has been designed to assess the above criteria and to embody best practice for marking Honours projects in a similar manner to that of other Scottish universities (e.g. Edinburgh, St. Andrews, Glasgow and Heriot-Watt).

What Markers Look For

It is very important that you document carefully anything which you brought into the project "from outside." This includes any external sources of information, and also specifically includes any work that has been done on the project by yourself or others prior to the month in which it started. The need for this is obvious - all students must be judged on the work done over the same period of time. There is no problem with a project which builds on previous work, provided the extent of that work is set out in some detail in the report. In general, when marking a project, staff are asked to look for and consider the following aspects:

The Marking Form

From 2015, we introduce a new project marking form, which your first and second markers fill in when marking your project. You will not normally be permitted to see the markers' comments on your project, but you may wish to use a copy of this form as your own check-list when writing up your project.

Your supervisor is not directly involved in marking your project. However, your supervisor will be asked to report on your project.

Your project markers will be given a copy of your supervisor's report just before they meet to agree a final mark.

Feedback

The Registry will provide you with a single overall mark for your project. Rather like final exams, this is the principal formal feedback mechanism. However, after your final results are published, you may request more detailed informal feedback from the Course Organiser.

Prizes and Awards

The department has a number of arrangements with external bodies which sponsor prizes and awards for student projects. After all projects have been assessed, the Departmental Exam Board will select which project(s) should be submitted to each of the schemes described below.

EDS Prize for Best Student in level 4

EDS sponsor a cash prize for the student with the best overall marks. The prize is normally presented by a representative from EDS at the Graduation Reception in July.

Amazon Prize for Best Honours Project

The best two projects are submitted to Amazon's Software development centre in Scotland. The winner is awarded a prize of an Amazon token.

Young Software Engineer Awards

Each year ScotlandIS invites every Scottish University Computing Science department to submit its best software engineering project to compete for a trophy and cash prizes. Prizes are also given for second and third places. In 2012 Steven Blake from the Department won third prize. The criteria used by the judges are:

- Level of Knowledge, Research, and Innovation
- Quality of Software Engineering, Presentation, Organisation, and Planning

Science, Engineering & Technology (SET) Student of the Year Awards

The SET Awards do not have a cash prize, but are highly prestigious. The criteria used by the judges are:

- Level of Achievement, Creativity, and Technical Depth
- Quality of Synopsis and Quality of Candidate

Please note: The student nominated for The ScotlandIS and SET awards will be asked to write a brief description of their project. For The SET award, the student's supervisor is asked to write a brief Citation for the student, and final judging (involving an interview) takes place on the day of the Dinner.

Intellectual Property Rights

By default, the current legal situation is that software that is written by undergraduates who are not employees of the University is treated in the same way as if they had written a novel at home in the evenings. In other words, by default, you would own the intellectual property rights (IPR) of your work. However, in practice, during your project you will typically be producing software as part of your regular daytime activity under the direction of a University supervisor, and using University facilities and resources. It is therefore necessary to establish clear title to any resulting IPR. Hence, since 1994, the University Court takes the act of your registration as a student to include the assignment of any IPR to the University. This is not so much that the University can get lots of extra income, which is unlikely. Rather, it is to help it defend its ability to do research. Research groups must be able to control the ownership of software they are developing, in order that future staff and students can continue to use and adapt it without threat of legal action or predatory pricing. It is also worth remembering that although your piece of software may seem very valuable to you, it usually only works as part of a larger system built by several people either as a team or in succession.

Note that you have not signed away your birth-right. The agreement only concerns IPR "arising in connection with or incidental to your course and studies here." You have not signed away rights to software developed at home for your own purposes on your PC. Instead you have entrusted the University with your claim to the IPR developed here in exactly the same way that University paid research and teaching staff have to do, so that any IPR can be safeguarded, and any financial reward shared out on a fair and equal basis.

In the case of students undertaking industrial projects as part of their course, we ask them to sign a form which also covers arrangements for examining reports and publication rights. The form assigns IPR to the University, and not the collaborating body. If the University makes money from the software, this will be shared equitably with you. For further information, contact the Department Industrial Liaison Officer.