

Individual Projects 2013/2014

Rules and Guidelines

Projects Committee

Dave Cohen D.Cohen@rhul.ac.uk, Carlos Matos Carlos.Matos@rhul.ac.uk

December 2, 2013

1 Introduction

This booklet concerns individual projects. It contains all the information that you will need from choosing a project title, right up to writing up, submitting and being assessed.

You should read this guide carefully in order to better understand the criteria used in assessing a project.

1.1 What is a project

An individual project is a piece of individual work done under the guidance of an academic supervisor. It should be about 200 hours of work (100 hours for a half unit project).

Different projects may stress theoretical aspects of a problem, or practical (programming/implementation) aspects, or may be essentially of a survey type.

- You will have to submit a project plan, programs that you have written, an interim report for the December Review meeting, draft and final project reports.
- You will also demonstrate the software that you have developed to staff, postgraduates, and fellow students.

1.2 Why do a project?

The Department **requires** all single honours Computer Science students to do a full unit individual project. Students on joint honours courses are also encouraged to do a full unit project, but other course commitments may mean that a half unit project is more appropriate. A half unit project is identical in structure to a full unit project. There is less time to complete a half unit of work so the choice of topics will be more limited.

A project is valuable to you because it is your opportunity to demonstrate your ability to work individually. *Success in your project will provide evidence of your skills to any future employer and students often use their project supervisors as referees after completing their degree.*

Your project should be the most enjoyable part of your undergraduate studies. You get to choose a topic of personal interest and to study it in depth.

1.3 How do you complete a project

You will be assigned to a supervisor. Your supervisor takes the role of your line manager in a company. They are certainly not there to teach you the material for your project.

You will have a brief meeting with your supervisor every week and at this meeting they will give you feedback on the work you have done, suggest new targets and help with any particular questions that you have.

You are expected to organise your own time, *keep a diary of what you have done*, deliver reports and programs for assessment and comment, and to update your svn repository regularly.

Use your supervisor to good effect. Hand in regular reports, show them your code and keep them up to date on your progress towards your final deliverables. They can help assess your writing style, your coding style and your understanding of the relevant theory.

The success of your project relies on steady hard work throughout the time you have available.

| Key Dates | |
|-------------------------------------|--------------------------------|
| First term | Date |
| Project Plan | 2pm on Fri, 4th Oct, 2013 |
| <i>Term 1</i> Reports | 2pm on Wed, 4th Dec 2013 |
| December Review Viva | Mon, 9th – Fri, 13th Dec, 2013 |
| Half Unit Project Submission | Date |
| <i>Half Unit</i> Draft report ready | Mon, 13th Jan, 2014 |
| <i>Half Unit</i> Final Report | 2pm on Wed, 29th Jan, 2014 |
| <i>Half Unit</i> Final Programs | 2pm on Wed, 29th Jan, 2014 |
| Full Unit Project Submission | Date |
| <i>Full Unit</i> Draft report ready | Fri, 21st Feb, 2014 |
| <i>Full Unit</i> Final Report | 2pm on Wed, 26th Mar, 2014 |
| <i>Full Unit</i> Final Programs | 2pm on Wed, 26th Mar, 2014 |
| Final Project Assessment | Date |
| Project Demo Days | Summer Term, 2014 |

2 Choosing a project and finding a supervisor

Projects are allocated in summer term before the academic year in which they are completed. This is to allow you to do some background research and begin a project plan over the summer vacation.

The list of available projects topics is published on the web page at [Topic List Web Site](#) early in the summer term. About a week after the publication of the list a [Chooser Web Site](#) is made available.

If you intend to do a project you must submit your preferences using the chooser web site. On this site you drag five project choices to a submission area, choose your name from a drop down list and press the Submit button. You then see a report of the projects you have chosen. You should make sure that the choices you submit are ordered according to how much you like them, with your favourite project topic at top of the list.

You can submit your choices any number of times, in case you change your mind, up to the submission deadline.

2.1 The project topics

As each supervisor has a limited capacity for project students, and some supervisors will only allow one student to do a particular project topic, it is imperative that you choose as wide a range of projects as possible on the chooser web site.

Some projects have *prerequisites* attached. Typically, this means that you must be registered for certain third year courses, or have certain key skills. Make sure that you pick projects that you will be able to do.

2.2 Designing your own project title

Students are usually *not* allowed to attempt non-standard projects (projects that are not on the departmental list). This is to ensure that all projects have a sufficient academic content, and that they are not too ambitious.

If you wish to attempt a project that is not on the departmental list of topics then you must find a supervisor willing to take on that project and submit the project description, including all of the sections that you see in standard topics list, to the projects committee. During the allocation process the projects committee will decide whether you are able to do the project that you have devised with the supervisor you have chosen.

In any event it is essential that you choose five topics from the departmental list using the project chooser web site.

2.3 Your supervisor

When all project preferences have been collected an optimising computer algorithm allocates students to appropriate supervisors, taking student preferences and staff workloads into account. You will be notified of your allocation by email and on the departmental web site.

You should arrange a first meeting with your supervisor before the end of the summer term. This is entirely your responsibility.

The project title allocated to you automatically by the system is not final. At your first meeting with your supervisor you might opt for a project better suited to your interests and skills. This must be a topic from the standard list that your supervisor is willing to supervise.

3 Organisation of the project

This section describes the process of doing an individual project.

3.1 Initial Meeting and Project Plan

After the project allocation is published and before the end of the summer term it is essential that you arrange an initial project meeting with your supervisor. At this meeting you should discuss the project topic and decide what background reading might be helpful over the summer.

You must also discuss the structure of your project. In what order will you be learning key skills and basic theory. You will also decide, in principle, what the final project will look like.

Following this meeting, over the summer, you will prepare a project plan:

- Briefly describe the project reports that you will be writing in the first term. Such reports will form the basis of your final project report. Typically you will write reports on key background concepts, mathematical theory, algorithms, technologies and relevant literature.
- Briefly describe “proof of concept” programs that you will write in the first term. These programs “prove” that you can code all of the algorithms required, that you can make an appropriate user interface, and that you can use key technologies and libraries effectively.
- Write a short “abstract” indicating why you are doing the project and what you expect the final project to achieve.
- Write a timeline that includes starting and delivery dates for all of the first term reports and programs. It should also include important milestones, with dates, for the second term.

Your project plan must be handed in to the office by 2pm on Fri, 4th Oct, 2013. It will be assessed and count 5% towards your final project grade.

Your supervisor will decide whether the first term schedule of reports and programs, together with the overall project milestones, is a sufficient and effective plan for the project. They will also help you to decide how best to modify and implement the plan.

3.2 Weekly meetings

It is vitally important that you attend a short weekly meeting with your supervisor, to discuss your project. These are the best way for both you and your supervisor to monitor the progress on your project.

These regular weekly meetings will only be twenty minutes long. As soon as possible after the start of the first term you should contact your supervisor to arrange the time for these meetings that fits in with your timetable and their other commitments.

During your first meeting slot you should discuss your project plan with your supervisor and agree the deliverables and milestones. You should decide on a schedule of work for the first two weeks so that you can start work on your project as soon as possible.

There will not be time in twenty minutes per week for supervisors to cover new material so it is important that you arrive each week prepared with questions and problems for discussion. You can ask your supervisor to read and review your work before the weekly meeting.

It is vital that both you and your supervisor keep records at *all* project meetings. In particular you *must* record any deadlines given to you and any deliverables required of you.

Bring a notebook to all meetings.

3.3 Project diary

It is important to keep a diary or workbook log of your project work. This will be invaluable when you come to write your final report as it will help you to remember problems that you found and dead ends that you investigated.

Use the notebook that you bring along to project meetings.

3.4 Your SVN repository

We also require you to keep all reports, programs, notes etc. on your SVN repository. This will mean that all of your work is backed up regularly and available to you and your supervisor from home or at the Department. *Use a sensible directory structure as you will have lots of files.*

It also provides a straightforward mechanism for submission of your final report and programs.

3.5 Supervisor's responsibilities

The supervisor is there to monitor and advise. They are not there to teach. There will not be time in twenty minutes per week for supervisors to teach new material. Supervisors will give you references and be willing to discuss problems *after you have studied* new material.

It is the supervisor's responsibility to attend each of the weekly meetings, or to re-organise the meetings if this is not possible.

The supervisor will keep an attendance register of your meetings, and of your general conduct during your project. This will count 5% towards your final assessment.

3.6 Paperwork

During the course of the project there is a minimum level of paperwork that must be maintained.

- There is a requirement for the **supervisor** to maintain a register and a performance record. This record will be assessed as 5% of the final assessment for the project. These records may be kept electronically.
- You **must** maintain a diary or log book of all your work. It is very useful when it comes to writing up as it is very hard to remember all that was done early in a project! Also, it can be used as a basis for discussions with the supervisor about problems and progress. You should get your supervisor to look at your diary periodically to make sure that you are keeping a log of all that you are doing.

4 December Review

1. In the penultimate week of term, (Mon, 9th – Fri, 13th Dec, 2013), a review will be conducted to determine your progress on the project to date.
2. A review panel will consist of three people: the supervisor, your project second marker, and one member of the projects committee.
3. At the review meeting you will give a ten minute presentation about your project.
4. The review panel will grade the presentation and the submitted material using the appropriate criteria (See Section 12.3).
5. The grade will be justified on a grading form with a short paragraph and feedback will be written by the supervisor and returned to you, by your supervisor, in a timely fashion.

See Section 12.3 for a detailed description of how your progress will be assessed at the December review.

4.1 Preparing for the December Review

Not only will we assess your presentation in December but we will also look at programs and reports that you written.

The programs that you write in the first term are normally “proof of concept” type programs. You will have seen new algorithms, new hardware, new library interfaces or even more complex data structures. It is best to experiment with these new concepts by coding them in small working programs. It is a good rule that we do not truly understand an algorithm until we have successfully made it work in a program. These initial programs make the coding of the final project deliverables much more straightforward. many of the hard issues have already been solved. New material that you have learnt and programs that you have produced will be assessed under the heading “Technical Achievements” and will count 10% towards your final project grade.

The reports that you write in the first term are to help you write your final project report. They will cover theoretical and practical aspects of your project work. Ideally they will already have been put together into a single document that will form the basis of your final project report.

5 The project reports

Your **final project report** is your most important deliverable, counting 30% towards your final project mark. A final project report is approximately 15,000 words and must include a word count. It is acceptable to have other material in appendixes.

Your **interim report** for the December Review meeting will count 10% . Even if it is a collection of reports, the total word count should be about 5,000 words. This should summarise the work you have done so far, with sections on the theory you have learnt and the code that you have written.

5.1 Writing the final report

You should aim to agree on the outline of your report with your supervisor as early as possible, normally at the start of the second term. This will allow you to **write up your work as you go along**. An example of a typical list of headings for a project report is attached.

As your project progresses keep together all the work you do (in your svn archive), including early incorrect ideas and program fragments. These will all be essential in explaining the development of your work.

Your supervisor **must** see a complete draft of your report by Fri, 21st Feb, 2014 (or Mon, 13th Jan, 2014 for Half Unit projects). Their feedback can help improve the final version.

5.2 What must your report contain

1. Your report must contain a section **motivating** the project and giving the original **project aims**.

This section must include a description of how you think that the work involved in your project will help in your future career.

2. Your report must contain a short section on **professional issues** (See Section 6) that raised concern during the year, particularly with respect to doing your project or the material contained in your project.
3. You must also have some sort of **self-evaluation** in the assessment section: How did the project go? Where next? What did you do right/wrong? What have you learnt about doing a project?
4. You must include a description of **how to run any software** that you have submitted, including any environmental requirements (Java version number, IOS version etc.,)
5. All projects will also need a **bibliography** of works referred to in the text, or that have been read in order to understand the project.

To avoid the accusation of plagiarism (See Section 8) you must cite anything that you quote from (or use images/diagrams or even précis/reword) in the text.

5.3 What should your report contain?

1. Some kind of **Introduction**. This should be broken into an **abstract** of the project, a section of **motivation**, and a list of **project goals**.

2. Some kind of **theory section**. This might include a **literature survey**, sections on **specific theory**, or even an **interesting discussion** on what you have achieved in a more global context.
3. You will need sections describing the **software engineering method** that you used. If your project is based on a **software product** then this may be most of your report.
4. If your project is mostly **theoretical** then you should include some **theory development**. This might include **small programs** to investigate certain things, **explanations of algorithms**, or even descriptions of the **particularly hard bits of theory**. A theoretical project will have **results** and some **analysis of results**. *Even short programs should be developed using appropriate software engineering methodology.*
5. If your project is **hardware oriented** then you will need to describe the hardware development process. The details of what sections should be in your report should be discussed with your supervisor.
6. Every report needs some kind of **assessment** section. This might include a **realisation** section describing your experience of the project, a **summary** of what you achieved, and a section on **problems and enhancements**. Of course you should have a short **conclusions** section summing up the whole process.
7. Lastly there are some added extras you might want to include. Perhaps parts of a **program listing**. Perhaps some **sample output** or **experimental results**.

6 Professional Issues in your final project report

Ethical behaviour is concerned with what is good or bad, with moral duty and obligation and as such deals with opinions and beliefs.

Professionalism in computing is concerned with the societal impact of computer technology and the creation and understanding of policies for the ethical use of such technologies.

Professional bodies such as the **British Computer Society** (BCS) and the **Association for Computing Machinery** (ACM) help ensure professionalism and ethical behaviour by providing standards and a code of individual conduct: guaranteeing certain levels of competence, integrity and a commitment to the interests of all end-users and other stakeholders.

I am amazed when I meet computer professionals in business and industry or even computer science teachers in colleges and universities who fail to recognise that their profession has social and ethical consequences *Terrell Ward Bynum (2003)*

After completing a Royal Holloway Computer Science degree we expect that you will be ready to be ethical computing professionals. To this end we include material on professional issues in our undergraduate modules.

The individual project is no exception. By completing an individual project, as well as the theory and practise essential to your chosen topic, you will have acquired skills in time management, prioritisation and both oral and written presentation.

Certainly you will have encountered some professional issues: correct citation, licensing, accessibility etc.,

We require that you complete a short section on professional issues in your final report.

What is required

The section in your project report must be clearly indicated. It can either be part of the general flow of the report or it can be an appendix. It must be approximately 1,000 words.

You must choose a topic that is relevant to your project (see the following section for examples). Then you could:

- describe an example from the public domain of what can happen when professional issues are not properly addressed; or
- write about how a particular issue has been of concern to you in your project; or
- describe some professional issue that has arisen during your project and discuss its ethical or practical importance.

This section must be reflective and thoughtful and is a requirement for a successful project submission. You **must** include a completed professional issues section in the draft report handed to your supervisor.

Possible Topics

Professional Issues occur wherever computing meets society. As such they are always concerned with how people interact with computers and software. This is a very wide area and you may well choose a topic not listed below but these are given as guidelines to help you.

- Usability - accessibility, replacing humans, artificial intelligence.
- Plagiarism - correct citation, using code with acknowledgement.
- Licensing - shareware, open source, copyright, patenting, reverse engineering.
- Safety - reliability, economic impact, trust, provided “as is” clauses.
- Privacy - web privacy, legal issues, data usage.
- Monopoly - proprietary formats, tie-ins, cartels, DRM, Google, Amazon, Microsoft, Apple
- Management - appropriate costing of time and resources at the start of a project. Revision during project. Consultation with stakeholders.

7 Programs and other technical material

As well as the project reports, you must submit any programs that you have written. These must be properly documented and well written.

The programs that you write in the first term will be submitted at the same time as the interim report, by 2pm on Wed, 4th Dec 2013. The mark given for technical achievement in the first term will be 10% towards your final project.

When you submit your final report you must submit finished programs for assessment. These will count 10% towards your final project.

You should discuss with your supervisor exactly what is required in addition to the report. For example:

- Long or complicated test output (referred to in the report).
- Examples showing the use of the project.
- A working program with instructions on executing it.
- Copies of papers and other reference material used for the project.

8 Plagiarism and acknowledgement of sources

Plagiarism (the unacknowledged use of other people's work) is a very serious offence and will be severely penalised. If you are in any doubt about what needs to be referenced and acknowledged, ask your supervisor for advice.

It is in the nature of a project that much of the material will not be original. You will have researched around your subject and discovered many sources of information. It is vital that any quote made from any source (including the web) should be properly acknowledged, both where it is used *within the report text*, and at the end of the report in the bibliography. Under no circumstances should copyrighted material be included in a project report without the proper permissions having been obtained, and any such inclusion should be agreed with your supervisor.

8.1 Bibliography

All sources of information which you use during your project must be listed in your bibliography. This includes books, articles, research papers, course notes and Internet sites. Quotations must be acknowledged, for example:

Henry Smith [1] states that "The problem of wild animals on campus can only be solved by the introduction of even wilder animals which will eat them." but the results of this project seem to contradict him.

In this case, the citation [1] would refer to a bibliography entry such as

1. Smith, H. *Modern University Life*. Wombat Press, 1997.

If you express someone else's idea in your own words, then you must also acknowledge their original expression of the idea. For example:

Smith [1] believes that an infestation of wild animals in a university can only be cured by introducing suitable predators. However, the obviously recursive nature of his proposal led me to consider more feasible alternatives.

8.2 Acknowledging Borrowed/Modified Examples or Theory

If you use examples from a lecture course or a book to illustrate your background theory, then you must acknowledge the original source. Similarly if you follow a book or lecture notes when presenting background theory, for example:

The following sequence of definitions is based on [1], with simplifications due to the fact that we are only considering finite widgets.

8.3 Other People's Code

There is no penalty for using other people's code in your project, **as long as you make it clear that this is code that you have not written**. However, it is expected that most of the code submitted with your project will be your own.

If any submitted program includes any pieces of code which you did not write yourself, then you must identify this code by commenting it, and say where the code came from. For example, if you copy an implementation of a particular algorithm from a book, you must

make it clear that you did not write those lines of code. Even if you modify someone else's code to use in your program you must make it clear that you began with someone else's code.

You should cite the original program code in your report, as well as clearly marking it with the original author's name in your own source code.

9 Submitting your work

This section tells you how to produce and submit your project plan, reports and programs. It also defines the circumstances in which the submission deadline can be extended.

No project submission is anonymous.

9.1 Submitting the plan and reports

In order to produce and submit any document (project report or plan), you need to do the following.

- Prepare the text with a word processor (for example Microsoft WordTM) or a document preparation system (for example, L^AT_EX), using font size no smaller than 10pt and standard or double line spacing. You need to leave a margin of at least 2cm on the left hand edge of the paper, to allow for binding. It is easiest to use the project templates given available on Moodle.
- Make sure that the front cover of the report contains the following information:
 - your name
 - your supervisor’s name
 - the project title
 - the year
 - either “Full unit project” or “Half unit project”, as appropriate

Your name and the year must also appear at the very top of the cover (this makes searching through the projects easier for us).

- Tag your project in your own svn archive. The tag should be named *PlanSubmission*, *InterimSubmission* or *FinalSubmission*. The tag must contain a subdirectory *Documents* containing **electronic copies of submitted documents, saved in Portable Document Format (pdf)**.
- Print two copies on a good quality printer—again, the document must be clearly readable. You can print it either single or double sided, as you prefer.
- If your document has more than ten sheets of paper then it must be bound with a plastic spine (available from the department office) and a transparent cover (also available from the office). Documents shorter than ten sheets can be stapled in the top left corner.
- Hand in **both printed copies of the report** to the departmental office by the submission date.

Remember that printers, binding machines and the office itself will be in great demand at the time of the deadline; try not to leave preparation and submission of your documents to the last minute.
- **With your final report you must also hand in a completed feedback form** (See Section 10).

9.2 Submitting your programs

You must make sure that it is possible for your project markers to run any programs which you have produced as part of your project. It is also essential that you not modify your programs after submission.

To this end you must create a tag in your own svn archive. The tag should be named *PlanSubmission*, *InterimSubmission* or *FinalSubmission* and have an appropriate subdirectory structure.

The tag must contain:

- A `README.txt` file describing the directory structure of your tag,
- A subdirectory *Documents* containing **electronic copies of submitted documents, saved in Portable Document Format (pdf)**,
- Source files for all programs,
- Any Makefiles, Ant files, XML etc.,
- Results files,
- Executable programs (if appropriate),
- A text file of instructions for December Review programs,
- A user manual (in pdf) for final programs,
- An installation manual (in pdf) for final programs.

Please use a sensible subdirectory structure.

10 Feedback, Monitoring and Complaints

When you submit your final project report, you must fill in the project feedback form (available from the project Moodle site) and hand this in along with your project report. The feedback will be used to inform future procedures and practices by the projects committee.

Your supervisor will monitor your performance during the project. This will count 5% towards the total project assessment.

Complaints

1. Students may complain about their supervisor only on grounds of insufficient monitoring, or inappropriate demands.
2. In the first instance a student should bring complaints to their supervisor.
3. If, after bringing a complaint to the supervisor, the student still has a grievance then they should bring their complaint to a member of the projects committee.
4. If the committee feels that the complaint is justified, and the student wishes, the projects committee will make representations to the supervisor.
5. If the committee feels that a complaint is justified then they will minute this and the grievance will be considered when assessing the project.

11 Marking Procedures and Extensions

Each project will be assessed by the *supervisor, a second marker and a member of the projects committee*. The assessments are used to judge the quality of the three marked components to a project: the project plan (See Section 12.2), the first term assessment (See Section 12.3) and the final assessment (See Section 12.4).

Markers will mark according to the marking criteria supplied and will justify marks and provide appropriate student feedback.

1. The project plan will be graded by the project supervisor.
2. The December review presentation will be jointly graded by the supervisor and a member of the projects committee.
3. The student's project performance mark will be determined by the project supervisor.
4. The project demonstration will be assessed by the second marker.
5. Project reports and programs (interim and final) will be marked independently by the project supervisor and a second marker.

Mark reconciliation

In the event of a small discrepancy between the independent marks for a report or program (less than 10% of the mark) the final mark will be the average of the two marks. Otherwise the following process will be followed:

1. The two markers will try to arrive at an agreed mark. The discussion towards agreeing a mark will be recorded.
2. In the event that agreement is not possible, a third independent marker will be assigned, and the projects committee will seek agreement amongst all markers as to the final outcome.
3. If no agreement is possible then the external examiner will be asked to adjudicate a final mark.

Extensions

If you require an extension then you must, in the first instance, contact your advisor. Extensions for projects will normally only be given for medical reasons.

No marking penalty will apply to a project submitted by a student on time or within the limits of an agreed extension. An extension can only be given by the academic advisor, with the agreement of the project supervisor and the projects committee.

An email confirming your extension must be emailed to the departmental administrator by

your advisor.

If you are not granted an extension and you hand in any project deliverable late then a standard marking penalty will be applied.

12 Guidelines for Assessment

This section is concerned with the assessment of a project.

Aims

The aim of the individual project is to give students the opportunity to complete a substantial piece of work. This involves organising their own time, deadlines and deliverables, and delivering a completed piece of work in a professional manner.

Objectives

The following objectives will be achieved by completing a individual project.

The student will be able to:

- work independently on a significant piece of work, organising deadlines and deliverables;
- learn new skills and theory from diverse information sources;
- make technical decisions after consideration of appropriate evidence and act on those decisions;
- present and discuss a technical subject;
- compose and complete a technical report;
- work steadily under guidance for the duration of a project;
- understand what is required of a computing professional.

12.1 Assessment Criteria

Whilst there are several different assessments made there are only three marked components to an individual project: the project plan (See Section 12.2), the first term assessment (See Section 12.3) and the final assessment (See Section 12.4).

We use a variety of assessment methods in order to be assured that each learning objective is properly assessed. For example the demonstration is where we assess your ability to describe and discuss the final deliverables of your project and any technical achievements that you have made.

Ultimately, the final mark of the project will be determined by the external examiners to ensure that the project marks are commensurate with individual projects across the university sector. Thus, the following criteria are guidelines for choosing the final classification of a project and have been constructed in consultation with our external examiners.

12.2 The Project Plan: 5%

The project plan must be printed and handed in to the Departmental Office by 2pm on Fri, 4th Oct, 2013.

The Project plan will count 5% towards the final grade for the project. The grade grid describes the score achieved for each level of attainment.

The plan should describe the following:

- abstract: an overview of the aims and objectives for the project;
- first term milestones and possible challenges;
- second term overview: key dates and main deliverables;
- bibliography and citation: a list of the sources (web pages, books, papers) that you read to help you decide on your plan;
- planning and time-scales.

| Grade Grid for Project Plan | | |
|-----------------------------|---|---------------------|
| Level | Description | Value |
| <i>Fail</i> | Little work. Poor writing. Abstract does not describe project aims. Few deliverables. Little evidence of thought on timeline. | Fewer than 25 marks |
| <i>Poor</i> | A poorly thought out plan, with few deliverables or a badly written abstract. Little evidence of thought on timeline. | 26-40 marks |
| <i>Basic</i> | Usually a well-written abstract. Deliverables are confused or dateline missing. Perhaps some thought on timeline. | 40-60 marks |
| <i>Good</i> | A project plan with a well written abstract and completed dateline, but with limited explanation of deliverables | 60-70 marks. |
| <i>Very Good</i> | A project plan with a well written abstract and completed dateline. Good motivation for, and explanation of, deliverables | 70-85 marks |
| <i>Excellent</i> | As <i>Very Good</i> , but with clear presentation and a wide range of milestones | More than 85 marks |

Normally, more than 90% of students score between Basic and Good and fewer than 5% of students achieve an Excellent mark.

You will receive written feedback on your plan from your supervisor, and a nominal grade (A-F) describing your performance.

12.3 The December Review : 25%

All reports that you wish to be considered for your December review must be printed and handed in to departmental Office by 2pm on Wed, 4th Dec 2013. Each separate report must be collated and stapled so that it can be easily read. The reports should be submitted in an appropriate folder, or bound together as a single document.

A single sheet of A4 giving the location, in your SVN archive, of code and examples for the review, must also be included.

The December Review will count 25% towards your final grade for the project. There will be three components adding up to this weight. The presentation will count 5% and will be graded by your supervisor and a member of the projects committee. The review reports will count 10% and be graded by your supervisor and an independent second marker. The proof of concept programs and other technical achievements will count 10% and be graded by your supervisor and an independent second marker.

You will normally have about 5,000 words in well written reports and at least 300 lines of effective working code submitted prior to the December Review. This should be enough to demonstrate your technical achievements.

The grade grids following describe the score achieved for each level of attainment.

12.3.1 December Review: Reports : 10%

These reports will count 10% towards the final grade for the project and be handed in by 2pm on Wed, 4th Dec 2013.

The (first term) written reports will be expected to contain the following contents:

- aims, objectives and literature survey;
- planning and time-scale;
- summary of completed work;
- bibliography and citations;
- some form of diary.

| December Review Grade Grid: Reports | | |
|-------------------------------------|--|---------------------|
| Level | Description | Value |
| <i>Fail</i> | Very little work. No evidence of intention to write a coherent final report. | Fewer than 25 marks |
| <i>Poor</i> | Poor writing, bad structure, too little material. Little evidence of thought. | 26-40 marks |
| <i>Basic</i> | Producing an adequate amount of material, dependent on your individual project, as discussed with your supervisor. Some evidence of software engineering. At least two reports on background theory. Clear presentation. | 40-60 marks |
| <i>Good</i> | As <i>Basic</i> . Good evidence of software engineering. At least two reports on background theory. Well written with citations. | 60-70 marks |
| <i>Very Good</i> | As <i>Good</i> . Also a clear outline of the final project report. Good use of images, tables etc., Descriptions of why the interim programs were written. | 70-85 marks |
| <i>Excellent</i> | As <i>Very Good</i> . Reports collated and contents page supplied. Clear evidence of achievement well beyond the norm. Looking towards a publishable final report | More than 85 marks |

Normally, more than 90% of students score between Basic and Good and fewer than 5% of students achieve an Excellent mark.

Your will receive written feedback on your first term reports from your supervisor, and a nominal grade (A-F) describing your performance.

12.3.2 December Review: Programs and Technical Achievement : 10%

The technical evaluation of your first term work will consider:

- demonstrate a practical understanding of material/theory/algorithms at a final year level;
- good description of the programs written or planned;
- good use of software engineering;
- demonstrate good programming practice;
- programs written should be clearly useful for completing the final project deliverable;
- programs written should work as designed and be simple to execute.

| December Review Grade Grid: Technical Evaluation | | |
|--|--|---------------------|
| Deliverable | Date | Weight |
| Level | Description | Value |
| <i>Fail</i> | Only poor quality code or none submitted. No new programming or algorithmic concepts/algorithms/data structures/use of libraries beyond second year level. | Fewer than 25 marks |
| <i>Poor</i> | Some working code without good documentation or poorly written. Weak evidence of engaging with the programming challenges of the project. | 26-40 marks |
| <i>Basic</i> | Working code, well written or adequately documented. Clear sense of purpose in programs. Evidence of software engineering. | 40-60 marks |
| <i>Good</i> | As <i>Basic</i> but also some innovation. Interesting algorithms coded, or perhaps use of complex library. Clear evidence of the design process and use of svn archive. | 60-70 marks |
| <i>Very Good</i> | As <i>Good</i> but with Clear focus on covering a wide range of topics necessary to complete the final programs. Final program initial design begun. | 70-85 marks |
| <i>Excellent</i> | As <i>Very Good</i> but also completing advanced targets from the project spec. or other significant extensions outside of the original project spec. Includes complex algorithms and programs researched and coded independently. | More than 85 marks |

Normally, more than 90% of students score between Basic and Good and fewer than 5% of students achieve an Excellent mark.

You will receive written feedback on your technical achievement from your supervisor, and a nominal grade (A-F) describing your performance.

12.3.3 December Review: Viva: 5%

Marks awarded by the supervisor and a member of the project committee for the ability of the student to defend their work.

The purpose of the project viva is to explore whether the student can:

- explain the aims and objectives clearly;
- explain the background/relevance/importance of the project and set it in the wider context;
- give a broad description of the project - i.e. how parts of the project fit together to form a coherent whole;
- briefly explain the theory underpinning the individual parts of the project (for example how algorithms work or which architectural options existed including their benefits/drawbacks);
- communicate well, supporting their work with a clear simple presentation.
- defend and justify decisions made during the project;

| December Review Grading Grid: Viva | | |
|------------------------------------|--|---------------------|
| Level | Description | Value |
| <i>Fail</i> | Does not understand the aims of the project let alone anything done towards achieving them. | Fewer than 25 marks |
| <i>Poor</i> | Does not understand basic theories relating to any part of the project and cannot defend any of it. | 26-40 marks |
| <i>Basic</i> | Understood much of what they have done. May be very hesitant on background theory. | 40-60 marks |
| <i>Good</i> | As <i>Basic</i> . Could not necessarily defend all decisions and maybe struggled if the conversation went beyond the scope of the minimum requirements for the project. | 60-70 marks |
| <i>Very Good</i> | As <i>Good</i> . The student clearly understood and defended nearly all aspects of the project and its background. Clear evidence of commitment to excellent performance on the project. | 70-85 marks |
| <i>Excellent</i> | As <i>Very Good</i> . The student clearly knew more than experts in the department about (some aspects of) the project and its context. | More than 85 marks |

Normally, more than 90% of students score between Basic and Good and fewer than 5% of students achieve an Excellent mark.

Your will receive written feedback on your viva from your supervisor, and a nominal grade (A-F) describing your performance.

12.4 The Final project submission : 70%

See Section 9 for more information on submission of your final project material.

Final project submission will not be marked unless two hard copies and an electronic copy correctly are submitted, together with a completed professional issues section (See Section 6), and a feedback form (See Section 10).

The assessments made after final project submission will count 70% toward the total grade for the project.

12.4.1 Final Project Submission: The Report: 30%

The marks for the final report will be divided as follows:

- Rationale (10%): Aims, objectives and a good introduction describing the structure of the report.
- Literature Review and Background Reading (15%): Description and critical analysis of relevant background material from books, research papers or the web. Analysis of existing systems that solve similar tasks;
- Contents and Knowledge (25%): Description of relevant theory - whether mathematical, algorithmic, hardware or software oriented. Also Adequate chapters on development and Software Engineering;
- Critical analysis and Discussion (15%): Clear evidence of reflection on the project process, its difficulties, successes and future enhancements. Any conclusions or results analysed or discussed appropriately;
- Structure and Presentation (20%): Good use of English. Clear and appropriate report structure. Nice use of figures;
- Bibliography and Citations (5%): Clear and appropriate bibliography with good citations. Must be clear and well formatted. See Section 8;
- Professional issues (10%): Should be a topic relevant to the project undertaken. See Section 6.

A full marking grid is given on the next page.

Final Project Assessment Grid: The Report

| | Marks | < 40% | 40-49% | 50-59% | 60-69% | > 69% |
|--|-------|--|---|---|---|--|
| Rationale | 0-10 | Problem statement or Introduction or Motivation missing or severely under-developed. Absence of focus. Tasks unclear or confused. | Marginal focus. Relevance of topic explained; problem statement poorly developed; | Good relevant introduction. Some shortcomings in clarity of purpose and associated objectives. | Clear Motivation and well-written focussed introduction that explains the structure of the report and the tasks to be done. | Clear statement of problem and associated objectives. Persuasive and comprehensive rationale. Tasks demanding knowledge clearly above second year level. |
| Literature Review and Background Reading | 0-15 | No attempt at critical comment; Serious gaps and omissions in literature. | Little attempt at critical comment. Large gaps and omissions. | Fair knowledge. Some gaps and omissions. Some attempt at critical comment. | Sound knowledge of background area. Some critical review. Good understanding. | Full critical review of literature relevant to study. Comprehensive knowledge |
| Contents and Knowledge | 0-25 | No evidence of understanding of the project area. Confused conceptual thinking. Poor description of the Software Engineering process | Little evidence of understanding of the project area. Conceptual framework incomplete. Inappropriate or poorly described Software Engineering processes | Evidence of understanding with clear explanations. Adequate coverage. Conceptual framework well developed. Nice clear description of the Software Engineering methodology used. | Good knowledgeable account of the project as titled. Ample coverage of the subject matter in sufficient technical detail. Clear explanation of Software Engineering techniques with motivation. | Excellent understanding and insight. Conceptual framework underpins study. Comprehensive expert account of topic. Well thought through Software Engineering content. |
| Critical analysis and discussion | 0-15 | Weak and unacceptable analysis; Inadequate use of evidence for discussion; No critical evaluation of results | Limited or logically inconsistent analysis. Superficial critical evaluation of results or value of evidence. | Appropriate critical analysis but limited. Clear presentation of findings. Good analysis of the project process. | Clear presentation of findings. Competent analysis. Evidence of ability to evaluate results. Conclusions justified appropriately. | High level critical analysis of the process and any deliverables. Clear understanding of the quality of the work. Nice conclusions. |
| Structure and presentation | 0-20 | Unacceptable layout in terms of structure and logical argument. Inappropriate use of English. Serious deficiencies in presentation. | Poor layout in terms of structure and logical argument. Poor literacy style and deficiencies in presentation. | Generally good layout and clear literacy style. Mainly appropriate presentation. Relevant use of chapter and section structure. Some appropriate figures or tables. | Correct, clear English. Clear and competent expression. Consistent layout of the project report. Clear overall structure. Good use of figures with clear referencing. | Excellent layout. Conforms to all technical specifications. Lucid style of expression in English. Appropriate and innovative presentation. |
| Bibliography and Citations | 0-5 | No bibliography | Bibliography present but poorly formatted or cited | Clear well written bibliography. Some citations. | Well formatted bibliography. Adequate citations at correct points in text. | Correct reference to sources and inclusion of a full bibliography. |
| Professional Issues | 0-10 | No real attempt to describe any professional issues | Professional issues addressed but poorly thought out and related to the project | Professional issues discussed that are shown to be relevant to the project | Clear discussion of professional issues. Well written and related to project material. | Thoughtful discussion of professional issues and how they have affected the project process. |

12.4.2 Final Project Submission: Technical Achievement: 20%

Marks awarded by the supervisor and the second marker together which reflect the candidates total technical achievement.

You will be evaluated on:

- Are there novel applications of standard ideas and techniques?
- Did the student work with techniques, software, theory or concepts which are clearly above second year level?
- Did the student master complex new technologies/platforms or have to study a significant body of literature from disparate sources to complete the project?
- Is the account in the final report technically accurate and concise?
- Are important (technical) decisions well made and argued? This includes good design decisions, choice or development of algorithms, scope of the project. It does not include choice of platform or development tools as these are not technical *achievements*.

| Technical Achievement Grading Grid | | |
|--|---|---------|
| Level | Description | Value |
| <i>Outstanding first class</i> (Award rarely.) | Definitely to be published in journal, patented or marketed. Clearly of a professional standard. | 95-100% |
| <i>Very good first class</i> (Typically award to 2% of students.) | Outstanding - suitable for publication or sale, with some small modifications or tidying up, in a suitable outlet. | 85-94% |
| <i>Good first class</i> (Typically award to 5% of students.) | Excellent definitely worth continuing. Clear evidence of potential as a product or publication | 75-84% |
| <i>First class</i> (Typically award to 13% of students.) | Substantial originality clear evidence of ability to work independently. | 70-75% |
| <i>Upper second class</i> (Typically award to 30% of students.) | Solid and competent piece of work, well executed, probably has some originality. | 60-69% |
| <i>Lower second class</i> (Typically award to 30% of students.) | Reasonable standard, shows some merit, probably lacks originality. | 50-59% |
| <i>Third class</i> (Typically award to 16% of students.) | Minimum acceptable, will demonstrate some effort but leaves a lot undone, does go very little beyond application of standard techniques. | 40-49% |
| <i>Borderline Fail</i> (Typically award to 3% of students.) | Unsatisfactory most of the promises in the specification are not properly fulfilled in the final product or report. Technically trivial, does not go beyond application of standard techniques, and might be even weak on this account. | 35-39% |
| <i>Fail</i> (Typically award to 1% of students.) | A start has been made, but virtually no meaningful results or design. | 20-34% |
| <i>Very bad fail</i> (Award rarely.) | There are some ideas of what project is about but no real work has been done. | 10 -19% |
| <i>Appalling</i> (Award rarely.) | There is a project title and description but not much else. | <10% |

12.4.3 Final Project Submission: End Product (usually software): 10%

Marks awarded by the project supervisor and 2nd marker for the *quality* of the end product.

- Does it work? Is it stable?
- Is the software usable? Is the interface appropriate for the application (a compiler might require more technical skills to run than an e-commerce site)?
- Does the code and system structure follow the design?
- How complete is the functionality with respect to the requirements?
- Is the coding clean and well documented?
- Does the SVN archive reflect the use of good software engineering principles including appropriate use of branches/tags?

| End Product Grading Grid | | |
|--------------------------|--|---------------------|
| Level | Description | Value |
| <i>Fail</i> | Nothing of any complexity has been developed. | Fewer than 20 marks |
| <i>Poor</i> | Virtually nothing there. Too little to even judge completeness, structure and usability. There is no realistic chance that the student can improve this to a pass standard. | 21-40 marks |
| <i>Basic</i> | Reasonable standard several minor and some larger problems. Functionality is somewhat complete and code and system structure are such that the system will be maintainable. Usability is somewhat hampered, more like a prototype. | 40-50 marks |
| <i>Good Enough</i> | Reasonable standard several minor and some larger problems. Functionality is somewhat complete and code and system structure are such that the system will be maintainable. Usability is somewhat hampered, more like a prototype. | 50-60 marks |
| <i>Good</i> | Good standard several minor problems. Functionality is mostly complete. Code and system structure are such that the system will be maintainable. Usability is somewhat hampered, more like a (very) good prototype. | 60-70 marks |
| <i>Very Good</i> | Excellent standard some very minor problems but quite usable. Functionality is mostly complete and code and system structure are such that the system will be reasonably maintainable. | 70-85 marks |
| <i>Excellent</i> | Professional standard could be shrinkwrapped and sold. Functionality is complete and code and system structure are such that the system will be highly maintainable. | More than 85 marks |

12.4.4 Effort and Organisation: 5%

Marks awarded by the supervisor for the ability of the student to plan and organise their project, as well as a reflection of the level of effort the student shows during the project.

You are evaluated on:

- the ability to arrange and attend supervisory meetings;
- the ability to alter their project plan and identify priorities as they arise
- the ability to keep an organised project diary
- the ability to maintain a consistently high level of effort throughout the project
- an assessment of whether the student can work independently or requires constant supervisory intervention
- communicate well and maintain a good professional working attitude towards the project

| Effort and Organisation Grading Grid | | |
|---|---|---------------------|
| Level | Description | Value |
| <i>Fail</i> | Student did not undertake any planning, and little to no effort was consistently spent on the project. | Fewer than 25 marks |
| <i>Poor</i> | The student showed little if any attempts at organisation. Despite prompting by supervisor the students effort was poor. | 26-40 marks |
| <i>Basic</i> | The student showed some organisational skills, and the level of effort in the project was generally adequate. Marks in the lower range are awarded if the student required regular prompting. | 40-60 marks |
| <i>Good</i> | The project was well organised and a significant amount of effort was usually given by the student at during the project. | 60-70 marks |
| <i>Very Good</i> | The project was well organised and a high level of effort was often given by the student during the project. | 70-85 marks |
| <i>Excellent</i> | The project was very clearly organised and the effort given was consistently of a professional standard. | More than 85 marks |

12.4.5 Project Demonstration: 5%

Marks awarded by the second marker for the ability of the student to defend the work.

The purpose of the project demonstration is to explore whether the student can:

- explain the aims and objectives clearly;
- demonstrate the working code;
- explain the background/relevance/importance of the project and set it in the wider context;
- briefly explain the theory underpinning the individual parts of the project (for example how algorithms work or which architectural options existed including their benefits/drawbacks);
- communicate well;
- support their work with a clear simple A3 poster.

| Project Demonstration Grading Grid | | |
|------------------------------------|--|---------------------|
| Level | Description | Value |
| <i>Fail</i> | Does not understand the aims of the project let alone anything done towards achieving them. | Fewer than 25 marks |
| <i>Poor</i> | Does not understand basic theories relating to any part of the project and cannot defend any of it. | 26-40 marks |
| <i>Basic</i> | Understood much of what they have done. The product works well enough. May be very hesitant on background theory. | 40-60 marks |
| <i>Good</i> | Could not necessarily defend all decisions and maybe struggled if the conversation went outside the scope of the project. Simple Poster on display. | 60-70 marks |
| <i>Very Good</i> | The student clearly understood and defended nearly all aspects of the project and its background. Clear evidence of potential. | 70-85 marks |
| <i>Excellent</i> | The student clearly knew more than experts in the department about (some aspects of) the project and its context. Project demonstration flawless. Good Poster. | More than 85 marks |