Individual Full Unit Projects 2019/2020 Rules and Guidelines

Projects Committee
Dave Cohen D.Cohen@rhul.ac.uk, Iddo Tzameret Iddo.Tzameret@rhul.ac.uk

March 25, 2019

1 Introduction

This booklet concerns individual projects, in particular for the following courses:

- CS3821 Full unit project (30 credits)

- CS3822 Individual project in artificial intelligence (30 credits)

- IY3821 Full unit project (Information Security) (30 credits)

CS3822 and IY3821 will involve projects relevant to Artificial Intelligence and Information Security respectively. CS3821 (full unit) will have projects drawn from a range of topics in Computer Science.

This document 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 300 hours of work.

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 have to present your project and demonstrate the software that you have developed to staff and fellow students.

- You will have to keep a work diary up to date, attend meetings for which you have prepared in advance, and store all of your work on the Departmental SVN server or on a private GitHub repository.

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 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 may 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. The project eventually is your own, and you have to take individual responsibility for its success.**

You will have a number of brief meetings with your supervisor and at these meetings 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 required to organise your own time, *to keep a work diary*, deliver reports and programs for assessment and comment, and to update your SVN or private GitHub repository [1] 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	
Choosing a Project	**Date**
Project list available	Friday, 15th March, 2019
System for submitting preferences becomes available	Friday, 26th April, 2019
Deadline for submitting preferences	Friday, 3rd May, 2019
Final allocation published	Friday, 10th May 2019
First term	**Date**
Project Plan	14:00 on Friday, 4th October, 2019
Interim Programs and Reports	14:00 on Friday, 6th December 2019
Interim Review Viva	Monday, 9th – Friday, 13th December, 2019
Full Unit Project Submission	**Date**
Full Unit Draft report ready (send by *email* to supervisor)	Friday, 28th February, 2020
Full Unit Final Programs and Report	14:00 on Friday, 27th March, 2020
Final Project Assessment	**Date**
Project Demo Days	Monday, 27th April, 2020

---

[1] When we refer to your SVN repository we mean your individual repository provided by the Department on `https://svn.cs.rhul.ac.uk`. When we refer to your private GitHub repository we mean a private educational repository owned by the department, to which we will give you access, not your own open repository (contact Dave Cohen to obtain one).

2 Choosing a project and finding a supervisor

Projects are allocated towards the end of the previous academic year. This allows you time to do some background research and begin a project plan over the summer vacation.

The list of available projects (excluding those from Information Security) topics is available on the Full Unit Topic List Web Site from Friday, 15th March, 2019. The list for Information Security students is available at Information Security Topic List Web Site.

On Friday, 26th April, 2019 a Full Unit Ballot Web Site is made available.

If you intend to do a project you must submit your preferences using the ballot 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 the top of the list.*

You can submit your choices any number of times, in case you change your mind, until the ballot deadline on Friday, 3rd May, 2019.

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 any particular project topic, it is imperative that you choose as wide a range of projects as possible on the ballot 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 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 ballot 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 must 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 week by week requirements and processes involved with 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 you will 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.

- List the material that you read while preparing your plan, describing its relevance and key content. This will typically include a main book, some web pages and perhaps some research or review papers.

- Write a short section describing the key risks (things that might go wrong) associated with your project. Each risk should have some mitigation: what you will do in order to make sure that the risk is not realised, or to contain the impact of the risk if it is realised.

Your project plan must be submitted by 14:00 on Friday, 4th October, 2019. It will be assessed by your project supervisor using the appropriate criteria (See Section 12.2), and will count 5% towards your final project grade.

Your supervisor will decide whether the 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. You will receive written comments from your supervisor to help you to improve your project plan.

3.2 Project meetings

During the first (autumn) term in which you are working on your project you are entitled to a twenty minute meeting with your supervisor in the first teaching week, and then a meeting once every two weeks. You should contact your supervisor as early as possible during this term to arrange the time for these meetings.

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

In the second term you will have a meeting with your supervisor in the first two weeks of term, and then three further meetings. The first of these second term meetings will be to review progress made so far, the outcome of the December review, and to plan the work for the term. You should arrange to meet again once in the middle of term to discuss any issues that have arisen. You should then meet twice more towards the end of the term to discuss your draft report, especially the professional issues section.

These meetings are the best way for both you and your supervisor to monitor your progress.

There will not be time in your project meetings for supervisors to cover much new material. So it is important that you arrive prepared with questions and problems for discussion. You can ask your supervisor to read and review your work before the project meeting. *Such work must be made sent to the supervisor for consideration at least two working days before the 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. You are be required to make a record of all project meetings in your project diary.

Bring a notebook to all meetings.

3.3 Project diary

You are required to keep a diary or workbook log. This will be invaluable when you 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 Saving to a file repository

We also require you to keep all reports, programs, notes etc. on your departmental SVN repository or in a private GitHub 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 structure as you will have lots of files.*

You will submit all work using the departmental anonymous submission system. For these submissions, your SVN or GitHub repository will also be very helpful. You can checkout a working copy, build appropriate executables etc., and then submit the resulting directory using the departmental anonymous marking system.

3.5 Supervisor's responsibilities

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

Your supervisor will keep an attendance register of your meetings, and of your professional conduct during your project. This, together with an assessment of your project diary, will count 5% towards your final assessment.

Your supervisor will also discuss relevant professional issues during your meetings in order to help you write the Professional Issues section of your report (See Section 6).

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

4   Interim Review

1. Towards the end of the first term, (Monday, 9th – Friday, 13th December, 2019), a review will be conducted to determine your progress on the project to date.

2. The review presentation will be assessed by a member of the projects committee (or possibly other staff member, but not your supervisor). You will give a ten minute presentation about your project which will be assessed using the appropriate criteria (See Section 12.3).

3. The submitted material, reports and programs, will be assessed by your supervisor and a second marker using the appropriate criteria (See Section 12.3).

4. Feedback on your performance will be returned to you in a timely fashion.

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

4.1   Preparing for the Interim Review

Not only will we assess your presentation, but we will also look at programs and reports that you have 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 novel and 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. You must combine them into a single document for submission. This report will count 10% towards your final project grade. The total word count should be about 5,000 words.

These interim reports 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 Interim Review meeting will count 10% . It should be submitted as a single combined document. The total word count should be about 5,000 words. The interim report 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. 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. We have also made available on Moodle a template final report both in Word and in LATEX format. All reports must be submitted *as a single file*, in Portable Document Format (pdf).

As your project progresses keep together all the work you do (in your SVN or private GitHub repository), including early incorrect ideas and program fragments. These will all be essential in explaining the development of your work. Also remember to keep your project work diary up to date with any work that you do for your report.

Your supervisor **must** see a complete draft of your report by Friday, 28th February, 2020. *This can be submitted by email as a Portable Document Format (pdf) file.* Their feedback can help improve the final version.

5.2 What must your final report contain

1. 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. 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. 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. A description of **how to run any software** that you have submitted, including any environmental requirements (Java version number, IOS version etc.,)

5. A **bibliography** of works referred to in the text, or that have been read in order to understand the project.

6. A **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.

7. Sections describing the **software engineering method** that you used. If your project is based on a **software product** then this may even be most of your report.

8. Lastly there are some added extras you might want to include. Perhaps parts of a **program listing**. Perhaps some **sample output** or **experimental results**. Often you will include a **user manual** (though complete installation and operating instructions are mandatory). These extra documents may be put into an appropriate appendix so as not to count towards the word limit.

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.

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 given 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, both for the Interim review and with your final report. These must be properly developed and documented and should be well written.

The programs that you write in the first term will be submitted at the same time as the interim report, by 14:00 on Friday, 6th December 2019. Your technical achievement in the first term will count 10% towards your final project.

When you submit your final report you must submit finished programs for assessment. These will count 30% towards your final project.

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

- Long or complicated test output (referred to in the report).

- Examples showing the use of the project.

- Detailed instructions for executing submitted programs.

- Copies of papers and other reference material used for the project.

It is usual to include such material in the appendix to the final report.

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 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 you 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.

Other people's code must come from publicly available, free for educational use, sources. It must not be commissioned, nor can it be the work of any colleague at RHUL.

9 Submitting your work

This section tells you how to produce and submit your work. The procedure for applying for a submission deadline extensions are described in Section 11. Extensions are normally only given for medical reasons.

Please note that there is no paper submission of any kind, except for the feedback form and the draft report, which are not graded. All graded submissions must be made using the departmental anonymous submission mechanism.

- **Use the departmental (anonymous) submission mechanism to submit your work.**

- Interim and Final reports, and your project plan. Each must be delivered *as a single file* in Portable Document Format (pdf) as described in Section 9.2.

- Programs and supporting material. See Section 9.3 for a description of what is required.

- **Please use a sensible subdirectory structure for your anonymous submission directory.** Your report must clearly document the structure of your submission directory.

- **You must also complete a feedback form. These will be collected at the demonstration sessions.** (See Section 10).

9.1 Submitting your draft report

In order to produce and submit the draft report, you need to do the following.

- Prepare the text with a word processor (for example Microsoft Word$^{\text{TM}}$) or a document preparation system (for example, LaTeX), using font size no smaller than 10pt and standard or double line spacing. It is safest to use the project templates available on Moodle.

- Your submitted draft report must be called `USERNAME.draft.pdf`

- Make sure that the front cover of your draft report contains the following information:
  - your name
  - your supervisor's name
  - the project title
  - the year
  - the words "Full unit project"
  - the course code (eg "CS3822 project in artificial intelligence")

  Your name and the year must also appear at the very top of the cover.

Your draft report can be printed and delivered directly to your supervisor (not the office), or you can email the file as an attachment to your supervisor.

9.2   Submitting the plan and reports

In order to produce and submit the interim report, the final project report and the project plan you need to do the following.

- Prepare the text with a word processor (for example Microsoft Word^{TM}) or a document preparation system (for example, LaTeX), using font size no smaller than 10pt and standard or double line spacing. It is safest to use the project templates available on Moodle.

- Your submitted plan must be called `USERNAME-plan.pdf`, your interim report must be called `USERNAME.interim.pdf` and your final report must be called `USERNAME.final.pdf`

- Make sure that the front cover of your report contains the following information:

   - your name
   - your supervisor's name
   - the project title
   - the year
   - the words "Full unit project"

   Your name and the year must also appear at the very top of the cover.

9.3   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.

For some submissions it has been hard in the past for graders to run students' code. Therefore, please make sure that you supply careful instructions for executing your code. In particular you *must indicate which packages might need to be installed* to make your code run, and provide detailed and correct instructions.

To assist in running and assessing your code you **must** provide screenshots in your report and *should*, where appropriate, include videos of the running software with your submission. Put any such videos on YouTube and provide links.

In particular, you must include in your submission:

- A `README.txt` file describing the directory structure of your submitted directory,

- 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 (these can also be added as an appendix of your final report),

- Executable programs (if appropriate),

- A text file of instructions for Interim Review programs (this may be included in the program documentation),

- A user manual (in pdf) for final programs (this may be an appendix of your final report),

- An installation manual (in pdf) for final programs (this may be an appendix of your final report),

- Screenshots of your program run,

- Where appropriate, include videos of the running software. Put any such videos on YouTube and provide links.

10   Feedback, Monitoring and Complaints

When you attend your final project demonstration, you will be required to return a completed project feedback form. Blank forms will be available at the demonstrations but it will save you time if you complete a form (available from the project Moodle site) ahead of time and bring it along to your project demonstration. The feedback will be used by the projects committee to inform future procedures and practices.

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 a 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 Interim review presentation will be graded by a member of the projects committee (or other staff member, but not your supervisor).

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 the second marker.

Mark reconciliation

In the event of a small discrepancy between the independent marks for a report or program (less than 10%) 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

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 granted using the standard College procedures. Please make your supervisor aware of your application for an extrension, and the outcome of the application

An email confirming your extension must be emailed to your supervisor.

If you hand in any project deliverable late (allowing for any agreed extension) then the 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 interim 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 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 submitted by 14:00 on Friday, 4th October, 2019.

The Project plan will count 5% towards the final grade for the project. The grade grid describes how we grade the project plan.

The plan should describe the following:

- abstract: an overview of the aims and objectives for the project;

- first and second term milestones, both reports and programs;

- risks and mitigations;

- bibliography: the sources that helped you decide on your plan;

- planning and time-scales.

Grade Grid for Project Plan			
	Marks	Description	Percentage
Abstract	30	Copied from the project list	< 40%
		Describes and motivates the final project	40–69%
		Describes and motivates the final project. Gives well thought out individual project goals	70–100%
Timeline	40	Milestones missing, poorly thought through or just copied from project list	< 40%
		Milestones are adequate or good, each with dates but some have insufficient motivation	40–69%
		Good list of well explained milestones	70–100%
Bibliography	15	Bibliography missing, unusable or just includes web pages	< 40%
		Clear use of more than one source. Not just web sites. Nice evidence of good research.	40-69%
		Bibliography items are very good. Each has a short relevant discussion of its value to the project	70–100%
Risk Assessment	15	General Risks just given such as "may get behind"	< 40%
		Risks associated with deliverables and show clear thought. Ideally each (or some) risks are provided with mitigations that seem reasonable	40–69%
		A good case is made for understanding how the project might fail to proceed and what should be done about it. Some risks have likelihoods and importance.	70–100%

You will receive written feedback on your plan from your supervisor.

12.3   The Interim Review : 25%

All reports that you wish to be considered for your interim review must be submitted by 14:00 on Friday, 6th December 2019. The reports should be submitted in an appropriate folder.

Your report must clearly document the structure of your submission directory, possibly in a short appendix.

The Interim 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 a member of the projects committee (or other staff member). 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 submitted prior to the Interim 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 Interim Review: Reports : 10%

These reports will count 10% towards the final grade for the project and be submitted by 14:00 on Friday, 6th December 2019.

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.

The total word count should be about 5,000 words.

Interim Review Grade Grid: Reports			
	Marks	Description	Percentage
Quality of Writing	20	Poor writing or structure or too little material	< 40%
		Well structured and good use of English	40–69%
		Organised and collated towards a final project report	70–100%
Background Theory	40	Very little theory or poor evidence of understanding	< 40%
		Evidence of understanding of background theory, ideally demonstrated by explanation of appropriate proof of concept programs	40–69%
		Clear evidence of independent thought using background theory imaginatively	70–100%
Project Diary	10	No diary or just a list of achievements	< 40%
		Diary of achievements well related to original plan	40–69%
		Reflective diary showing strong evidence of analysis of process	70–100%
Structure	15	No bibliography and poor references to the literature	< 40%
		Interesting and appropriate bibliography or many appropriate well understood references	40–69%
		Excellent evidence of background reading and appropriate use of references in the report	70–100%
Software Engineering	15	Poor evidence of use of SE tools and methodology	< 40%
		Clear evidence of an engineering approach to developing software, including appropriate use of TDD and UML	40–69%
		Excellent SE. Clear evidence of all appropriate processes including the use of a revision control system and a good test strategy	70–100%

You will receive written feedback on your first term reports from your supervisor.

12.3.2 Interim Review: Programs and Technical Achievement : 10%

The technical achievement grade for your first term work will consider:

- demonstration of a practical understanding of material/theory/algorithms at an appropriate level;

- a good description of the programs written and planned;

- demonstrating good programming practice;

- that programs written are clearly useful for completing the final project deliverable;

- that programs written work as designed and are simple to execute.

Interim Review Grade Grid: Technical Evaluation		
Level	Description	Value
Inadequate	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
Poor	Some working code without good documentation or poorly written. Weak evidence of engaging with the programming challenges of the project.	26-39 marks
Basic	Working code, well written or adequately documented. Clear sense of purpose in programs.	40-59 marks
Good	As *Basic*. Interesting or complex algorithms coded, or perhaps the use of complex library. Clear evidence of a design process.	60-69 marks
Very Good	As *Good* but with clear focus on covering a wide range of topics necessary to complete the final programs. Also final program initial design begun.	70-85 marks
Excellent	As *Very Good* but also completing advanced targets from the project specification or other significant extensions outside of the original project specification. A nearly complete good quality project.	More than 85 marks

You will receive written feedback on your technical achievement from your supervisor.

12.3.3 Interim Review: Viva: 5%

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

The purpose of the project presentation 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 that keeps to the allowed time.

- defend and justify decisions made during the project;

Interim Review Grading Grid: Presentation		
Level	Description	Value
Inadequate	Does not understand the aims of the project let alone anything done towards achieving them.	Fewer than 25 marks
Poor	Does not understand basic theories relating to any part of the project and cannot defend any of it.	26-39 marks
Basic	Understood much of what they have done. May be very hesitant on background theory.	40-59 marks
Good	As *Basic*. Could not necessarily defend all decisions and maybe struggled if the conversation went beyond the scope of the minimum requirements for the project.	60-69 marks
Very Good	As *Good*. 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
Excellent	As *Very Good*. 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.

You will receive written feedback on your presentation from your supervisor.

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 an electronic copy is correctly submitted, including a professional issues section (See Section 6).

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 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 (20%): Description of relevant theory - whether mathematical, algorithmic, hardware or software oriented. Also adequate chapters on development and Software Engineering;

- Critical analysis and Discussion (10%): A discussion of actual project achievements and how successful the project was. Clear evidence of reflection on the project process, its difficulties, successes and future enhancements. Any conclusions or results analysed or discussed appropriately;

- Technical Decision Making (10%): Are important (technical) decisions well made and argued? This includes good design decisions, choice or development of algorithms, scope of the project.

- 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.

A final project report is approximately 15,000 words and must include a word count. *It is acceptable to have other material in appendixes.*

The section on professional issues should be about 1,000 words.

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 underdeveloped. 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. Some tasks 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	0-20	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. Expert account of topic. Well thought through Software Engineering content.
Technical decision making.	0-10	There is no evidence of thought behind choices made.	Design Decisions and choice of algorithms justified.	Alternatives considered for important technical decisions such as the use of patterns, algorithms or architecture.	Clear presentation of technical decisions suggesting that a careful approach has been taken.	Technical descions have ben made and presneted in way that demonstrates excellent understanding.
Critical analysis and discussion	0-10	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 literary style and deficiencies in presentation.	Generally good layout and clear literary 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.	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. 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: End Product (usually software): 30%

Marks awarded by the project supervisor and second marker for the *quality* of the end product and the candidate's total technical achievement.

In the unlikely event that the project does not involve the creation of a significant piece of software then an appropriate scheme will be used to allocate an end product grade to other forms of deliverable such as hardware.

If there is no evidence of frequent updates to your SVN or private GitHub repository of program code then a mark of zero will be awarded for your 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 or GitHub repository reflect the use of good software engineering principles including appropriate use of branches/tags?

- Does the product require techniques, tools, theory or concepts which are clearly above second year level?

- Does the product use complex new technologies/platforms or require the study of a significant body of literature from disparate sources to complete?

End Product Grading Grid				
	Marks	Description		Percentage
Code Quality	10	Code may not even be readable		< 40%
		Code meets a clean standard - possibly Checkstyle		40–69%
		Beautiful code, nicely commented, clearly structured		70–100%
Structure and Design	15	Code may not even be broken up according to functionality		< 40%
		Clear evidence of a design process. UML and design patterns used.		40–69%
		Design process shows refactoring. Evidence of careful use of appropriate data structures.		70–100%
Functionality	10	Code may not even work as expected or not possible to see code running using supplied instructions.		< 40%
		Code has all functionality as specified in the original project specification		40–69%
		Functionality significantly exceeds original specification. Clear and effective instructions, both for usage and installation.		70–100%
Usability	5	User interface unclear missing, or inconsistent		< 40%
		User interface effective and appropriate. Code works well in its designed environment (resizes or switches orientation or deals properly with life cycle)		40–69%
		Clear evidence of user interface design and thought about the user experience. Some attention on accessibility, or other enhancements		70–100%
Stability	10	Code has no documentation or overall structure is very weak		< 40%
		Code has good documentation and includes clear use of regression testing		40–69%
		Excellent SE. Clear evidence of all appropriate processes including the use of braches and merging in a revision control system and a good test strategy		70–100%
Technical Expertise	30	Project as delivered requires no final year expertise		< 40%
		Project requires techniques, tools, theory or concepts which are clearly above second year level		40–69%
		Some tools, theory or techniques employed show evidence of mastery at a high level		70–100%
Complexity	20	Code or results are very simple or do not represent what would normally be expected for this project		< 40%
		The project as delivered required learning of a significant amount of material, or the use of complex technologies		40–69%
		The project involved a deep understanding of several new areas of work and this is demonstrated in a successful implementation or clear and professional presentation of theory		70–100%

12.4.3 Professionalism: 5%

Marks awarded by the supervisor for the ability of the student to plan and organise their project as a professional. This score reflects the students ability to recognise and follow appropriate professional behaviour.

If there is no evidence of frequent, reflective updates to your work diary then a mark of zero will be awarded for your professionalism.

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, understanding relevant professional issues, and maintain a good professional working attitude towards the project

Effort and Organisation Grading Grid		
Level	Description	Value
Inadequate	Student did not undertake any planning, and little to no effort was spent on the project.	Fewer than 25 marks
Poor	The student showed little if any attempt at organisation. Despite prompting by supervisor the students effort was poor.	26-39 marks
Basic	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.	40-59 marks
Good	The project was well organised and a significant amount of effort was put into the project by the student at during the project.	60-69 marks
Very Good	The project was well organised. The behaviour of the student achieved a high standard of professionalism.	70-85 marks
Excellent	The project was very clearly organised, taking into account wider professional issues, and the student behaved throughout in a professional manner.	More than 85 marks

12.4.4 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
Inadequate	Does not understand the aims of the project let alone anything done towards achieving them.	Fewer than 25 marks
Poor	Does not understand basic theories relating to any part of the project and cannot defend any of it.	26-39 marks
Basic	Understood much of what they have done. The product works well enough. May be very hesitant on background theory.	40-59 marks
Good	Could not necessarily defend all decisions and maybe struggled if the conversation went outside the scope of the project. Simple Poster on display.	60-69 marks
Very Good	The student clearly understood and defended nearly all aspects of the project and its background. Clear evidence of potential.	70-85 marks
Excellent	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