
CO3015 The Computer Science Project

Study Guide 2019/2020
School of Informatics

University of Leicester, UK

Contents

1	Module Overview and Learning Outcomes	1
1.1	What is an Individual Project?	1
1.2	Learning Outcomes	1
2	Deliverables: An Overview and a Time-Line	2
2.1	Project Deadlines.....	2
3	Deliverables: A Detailed Description.....	3
3.1	The Contents of Deliverables.....	3
3.2	Technical Achievement.....	8
3.3	The Format of Deliverables	8
4	Module Organisation	10
4.1	Lectures and Laboratories.....	10
4.2	Administration and Communication.....	10
4.3	Allocation of Supervisor, Topic, and Second Marker.....	10
4.4	Role of Supervisor	11
4.5	The Student Diary	11
4.6	Progression Monitoring using SVN Version Control.....	11
5	An In-Depth Overview of The Computer Science Project.....	12
5.1	General Points	12
5.2	Specific Points for Regular Computer Science Project	13
5.3	Accreditation Requirements for Computer Science	13
5.4	Sample Illustrations of the Content of Computer Science Projects.....	14
6	Feedback.....	15
7	Students with Specific Learning Difficulties and Long Term Conditions.....	16
8	Late Submission of Project Coursework.....	16
9	Plagiarism	17

1 Module Overview and Learning Outcomes

1.1 What is an Individual Project?

A project is comprised of

- a typed dissertation of around 10,000 words about a subject, which may be new to you or perhaps extends your knowledge of a subject you have already learned; and which describes a computational problem and its solution; and
- a software system that implements this solution.

Most projects begin with the formulation of a problem which can be solved by the development of some software. Next comes the study of background material (which might be theoretical, or practical, or both) which underpins the problem. Such background will come from research/scholarship, such as reading and assimilating books and web pages (and perhaps journal or conference papers), and studying existing software systems and components which are related to the project work. This will be followed by specification, design, implementation and testing of some software. A good project should consist of foundational ideas and theories that underpin the software, and details of the process that is followed in its production, leading to a dissertation and the software system.

During the Project you will utilise skills acquired in the other modules. You will also assimilate information from a variety of new sources and demonstrate the ability to pursue independent study.

Information that is specific to this type of project can be found in Section 5. You may read the earlier sections first to get a broad understanding of the contents of the project modules.

Note: as a module with a 100% coursework assessment, resits for this module will not normally be granted.

1.2 Learning Outcomes

After completion of this module, you should have the knowledge and skills to

1. Plan a development of a (medium scale) software system which is a solution to a (practical) problem.
2. Undertake well organised surveys of information sources.
3. Assimilate substantial background information, recognizing the key sources.
4. Undertake substantial software engineering developments as outlined in your plan.
5. Give oral presentations of your work.
6. Write up a detailed account of such work, and the final results, in a well organised report (dissertation).
7. Know how to describe, justify and defend such work, and to critically appraise what you have done.

8. Produce a portfolio of activities recording valuable skills, which you have gained throughout the project work and which you can demonstrate to future employers.

2 Deliverables: An Overview and a Time-Line

Your project provides you with an ideal opportunity to gain skills and experience that will be useful to you in your career beyond graduation. You must show that you can plan and carry out your work within specified deadlines. In order to complete this module, you must produce some *deliverables* that will be assessed. These deliverables consist of:-

Plan (Pass/Fail): A short document that outlines your aims, objectives and the requirements for your software. You also include a Gantt chart for your year's work.

Prototype Demonstration (3% of final mark): Demonstration of a simple piece of software that is an early release of your final software system, built to test basic concepts and processes, from which you will have a clear understanding of how to proceed to the final software.

Career Plan (Pass/Fail): A summary of your career plans.

Interim Report (7% of final mark): A report that outlines your basic software architecture, and your important algorithms and datastructures. It also includes a description of your prototype.

Interview (5% of final mark): An interview with your **second marker** prepares you for your mini viva (oral exam).

Dissertation (25% of final mark): A typed document summarising your year's work.

Software System (15% of final mark): A substantial piece of software that meets the requirements set out in your plan.

Mini Viva with Presentation (10% of final mark): An oral examination run by **your supervisor and the second marker**. It begins with a short presentation of your work, including slides.

There is also a mark of **30%** for **Technical Achievement**, explained in Section 3.2, and **5%** for **Effort**.

2.1 Project Deadlines

A guideline submission timetable follows and is correct at the time of production of this document. Any changes to this timetable will be displayed on the Blackboard.

Term One (23 September - 13 December 2019)

- **Plan:** 4pm, Thursday 17th October 2019, see Section 3.1.1.
- **Prototype Demonstration:** 25th November – 29th November 2019, see Section 3.1.2.
- **Interim Report and Career Plan:** 4pm, Thursday 5th December 2019, see Section 3.1.3.

Term Two (6 January - 27 March 2020)

- **Interview:** 24th February – 28th February 2020, see Section 3.1.4

Term Three (27 April - 5 June 2020)

- **Dissertation:** 4pm, Thursday 30th April 2020, see Section 3.1.5.
- **Software System:** 4pm, Thursday 30 April 2020, see Section 3.1.6.
- **Mini Viva with Presentation:** during the midsummer examination period (5 May – 22 May 2020), see Section 3.1.7.

3 Deliverables: A Detailed Description

3.1 The Contents of Deliverables

Note: Here is a summary of the description of the deliverables to be handed in during this project. The material described below should appear in your documents, but **you do not have to have section titles which match those of the list items below**. You may care to discuss more appropriate section titles with your supervisor.

3.1.1 Plan The Plan should outline the aims, objectives and detailed requirements for your project. It should be about 1,500 words in length, and typically around five to seven pages long (these are rough guidelines). The assessment of the Project Plan will take account of the items listed below.

Aims and Objectives Your Plan should have an introduction, which contains your aims and objectives.

Aims describe the broad purposes of the project, and state in general terms what you are setting out to do.

Objectives describe concretely how the aims are going to be met; it should be possible to assess concretely/quantitatively whether or not the (list of) project objectives have been achieved.

Note that the objectives should focus on your own actual work, and not the assessment deliverables! For example, an objective might be to produce a short written account of optimising parallel processors, or to present structured requirements for a hotel management system, but *must not* state “produce an interim report” as a project objective (but such details may form part of Planning and Timescales).

As part of your aims and objectives, you should give a *summary of the challenges and originality that you intend to bring to your project*, and outline the problem that you are trying to solve. You should explain why your work is more than a re-hashing of other peoples’ ideas, and what fresh input you are going to deliver.

Structured Requirements Your plan must also include a detailed software requirement for your project, which magnifies your objectives and provides additional details. You may present your detailed requirements in any way you see fit, but writing a *structured requirements list* is a good way of proceeding. This list should specify the *functional requirements [F]*, *resources [R]* and the *quality attributes [Q]* of your software. The aim is to demonstrate that you have a very clear understanding of *what* your final software must be able to do.

Planning and Timescales should detail your year’s work plan, and explain target dates and amounts of times required for completions of components of the project. At this stage,

a detailed plan should be given for Semester 1, and a sketch for Semester 2. Include a Gantt chart. You might consider specifying “milestones”, making clear what the key goals of your project are.

Submission This handin is due 4:00pm **Thursday 17 October 2019** and should be submitted via SVN in the directory /docs/1_project_plan.

3.1.2 Prototype Demonstration During the project supervision meeting during the week **25th November – 29th November 2019**, you should provide a full software demonstration of your current software prototype. While the prototype does not equate to a fully functional finished product it should highlight all work done so far in terms of implementing the software solution. This handin should be submitted via SVN in the directory /code/prototype_demo

3.1.3 Interim Report This will be about 3,500 words in length, and take account of the following. It is due at the end of Term 1.

Aims, Objectives Include original text or modify it if there are any changes to your original aims and objectives.

Survey of Literature/Information Sources Give a summary of the background research you have undertaken in order to carry out your project. List your information sources; these can be a textbook, a website, a published paper, etc. Provide a brief comment on your major information sources.

Description of Prototype You should write a short account of your prototype, outlining its software architecture, and explaining the key details of its implementation. Make clear what aspects of the prototype must be *extended* in the final system, and what parts of the final system will be additional to those in the prototype.

Software Architecture, Algorithms and DataStructures For the final software system you should give a high-level overview of the architecture of the system, explaining what the key components are, and briefly outlining their main functions. If your system is making use of important algorithms and data-structures, then these should also be described. The idea is that you demonstrate that you have a clear idea of the most important ingredients that will come together to form the basis for your final software system, and that you can give some technical details of these ingredients.

Planning and Timescales Outline any changes to your original plans if there are any; if not include original text.

Bibliography and Citations Marks will be awarded for the inclusion of a well organized and detailed bibliography, and the appropriate citation of the listed references throughout the Plan. *It is imperative that you cite all of your data sources, especially those from the Internet.* Further guidelines about the organization of your bibliography can be found in subsection 7.

Submission This handin is due 4pm **Thursday 5th December 2019** and should be submitted via SVN in the directory /docs/2_interim_report

In addition, you should

1. include your **diaries**, as explained in Section 4.5, on SVN in the directory /docs/0_diaries.
2. produce a **career plan**. In your project, you have the opportunity to gain skills and experience that will be useful to you well beyond graduation. Career planning is meant to help you to make the best possible use of the project experience for your career development. First, you will attend a compulsory careers presentation. This presentation will also introduce you to the concept of a career plan, how to prepare one and what to put on it. You must then complete your own career plan, including a reflection on how your project can contribute to your future career, and discuss the completed plan with your supervisor in a supervision meeting. A career plan template will be provided on the Blackboard for this module.

A copy of the career plan must be submitted through your SVN directory. You are encouraged to arrange a follow-up meeting with a Careers Advisor in March to discuss your career planning progress.

3.1.4 Interview You will be assessed by an *Interview* with your **second marker**, during the week of (**24 February – 28 February 2019**). Regular supervision will not take place during this week. The interview will last for about twenty minutes. You will need to give a demonstration of your software and will be asked questions about all aspects of your project work; the questions will be both routine and more challenging, and may range from simple ones to quite technical ones. Note that you should not necessarily expect to be able to answer all of the questions, but supervisors will be looking for both general understanding, and innovative and intelligent replies. The Interview will help you prepare for your mini viva and allow you to meet your second marker. **You are responsible for arranging the interview with your second marker.**

3.1.5 Dissertation The Dissertation is a complete report of the year's work. It should take account of the components listed below. A typical report will be about 10,000 words in length (though reports can be longer or shorter). You should strive to use the minimum number of words to maximum effect. The Dissertation is due in at the start of Term 3 and should contain the following:

Abstract This is a short (approximately 200-300 words) but a very important section that *summarises the entire report*. Readers should be able to become acquainted with the entirety of your work from this section. It should, therefore, contain a very brief statement of the problem, background information, what you did and what were the main conclusions of the project. It may be useful to write this section last, but it should be the first section in your report.

Introduction Provide an introduction to your project, setting it in context and stating briefly what the aims and objectives you are setting out to achieve are.

Survey of Literature/Information Sources Summarize the research and scholarship you have undertaken in a separate "Background Research" section or chapter or, if you prefer, background material can be integrated throughout the other chapters of the project. Discuss with your supervisor.

Requirements You should formalize and state the requirements of your project. Ensure that you can measure whether you have achieved these requirements later.

Specification/Design/Data-Structures/Algorithms student should agree with the supervisor which of these areas should be included. All projects must include an implementation and thus the dissertation should outline the design process and the rationale behind any decisions made during the design process.

Implementation+Testing You should give a written account of the Software System code, outlining the top-level structure of the system, and explaining how the crucial sections of the code were implemented, and how they work. You may wish to include small samples of code in an appendix if necessary.

Critical Appraisal Conclude your project with

1. a summary and critical analysis of the work completed;
2. a discussion of the social, sustainability, commercial and economic context (or academic context if relevant);
3. a short assessment of your own personal development during the project.

Under (1), evaluate your project results in view of the original aims. What went well, what went badly, and what would you do differently with hindsight?

Under (2), explain the potential impact on people or businesses from your project, or from similar projects. What are potential benefits or risks to society? Are your methods and technologies used in the wider economy? Can they be used directly for commercial purposes, or suitably adapted? Is your project relevant only in an academic or teaching environment?

Under (3), briefly describe the knowledge and skills that you gained while developing the project, both technical and otherwise. How do you judge their relevance for your own future career?

Note that the appraisal, covering (1) and (2) should be quite detailed, and students should try to give a reasonably penetrating and critical analysis. Superficial appraisals will not attract many marks.

Conclusion An overall statement summarizing the entire project and stating whether or not you have met the project aims and objectives.

Bibliography and Citations List all relevant sources of information clearly. *At this stage, the bibliography may well have been expanded from that in the Interim Report.* **IT IS IMPERATIVE THAT YOU CITE ALL OF YOUR DATA SOURCES, ESPECIALLY THOSE FROM THE INTERNET. THIS MUST BE MADE CLEAR THROUGHOUT THE PROJECT REPORT.**

Submission This handin is due **4pm Thursday 30th April 2020** and should be submitted via SVN in the directory /docs/3_dissertation.

The Dissertation should be structured into chapters. The titles of the chapters may, but do not have to, correspond to the list above. Usually, good projects will have their own unique set of chapter titles which suitably reflect the year's work.

You should write about all of your work in detail, so that a knowledgeable computer scientist can read your report and understand what you have, perhaps leaving them with a few questions—you are not writing a textbook, but you must be as clear and lucid as possible.

You should also remember that the report may be read by *some* markers/examiners who may not have access to your software. *You must, therefore, include enough illustrative material such as screen shots to enable the marker to form an impression of your software.*

3.1.6 Software System You should submit all of the code necessary for compiling and running the project. This includes all of the code you have written, and any third party software components. You should also include an executable. There **must be a README file, which lists all of the files, explains how a user can install the software from the original code, and how the executable itself can be activated. Please give details of the operating system required, and any other programs that are required.**

Note that usually no changes may be made in the software between the submission of the Dissertation and the mini viva. If you have a legitimate reason for making changes (for example if a disastrous bug is discovered) then a formal request for permission to make changes must be made in writing to your supervisor (with a copy to the module convener), who may then give written permission. *In such cases, you must outline during the mini viva how your code differs from that submitted, by showing the examiners what code has changed.* This handin is due by **4pm Thursday 30 April 2020** and should be submitted via SVN as a subdirectory of the directory `/code/`.

3.1.7 Mini viva A *mini viva* (oral exam) will take place during the **midsummer examination period (5 May – 22 May 2020)**. This will last for 30 minutes. You will be notified of the time of your mini viva, by email, a week or so before it is due to take place.

Your mini viva should begin with a short presentation (no more than 10 minutes) using PowerPoint or an alternative technology. Remember that if you do take two minutes to explain each slide you will only be able to get through five slides. Realistically the presentation cannot do more than outline the general idea of the project and any points of particular interest. The presentation should briefly outline the motivation and aims of the project while identifying the main technical and theoretical challenges. It should be clear from the presentation what has been achieved during your project.

Typically the first and second markers will interview you for about 20 minutes, beginning after your presentation, and you will also demonstrate your software. 10% of the mark for CO3015 will be allocated for the mini viva, and information from this session will also be used in assessing the Dissertation and Software System. Questions will be asked at all stages of the mini viva, and, just as in the interview, you will be examined thoroughly and rigorously on all aspects of the project. You should not expect to be able to answer all questions in depth, but should be able to provide sensible and intelligent responses. Supervisors will wish to establish that your work is indeed your own. If the code has been written by you, you should be able to explain it in detail. If the code in question belongs to a 3rd party component, you should demonstrate clearly that this has been documented, and also be able to outline the broad functionality of the component and how it fits into your overall software system.

Note: the mini viva is an important part of the project assessment process, and you cannot graduate without attending. It is your responsibility to ensure that you attend your mini viva, and if you should miss it, to contact your supervisor and re-arrange it.

3.2 Technical Achievement

Marks of up to a total of 30% will be awarded for technical achievement throughout your year's work, and in particular this will be judged through your Dissertation and Software System. Credit will be gained from novel applications of standard ideas and techniques; working with concepts, ideas, software, theory (and so on) which is clearly at or above third year level; concise writing which is technically detailed; informed decision making (e.g. including a sensible choice of good, detailed, Use Cases which inform the coding, rather than pages of trivia); the use and description of sound mathematical techniques where appropriate.

3.3 The Format of Deliverables

1. All documents must be A4 paper sized. Characters must be not less than 10pt.
2. All documents must have numbered pages. The first page of the main body of text should be number with the arabic numeral 1. Preliminary pages such as the table of contents should be numbered using small roman numerals (such as i, ii).
3. The Plan, the Interim Report, and the Dissertation must start with a title page supplying *at least* the following information:
 - (a) "CO3015 Computer Science Project" **or** "CO3016 Computing Project" **or** "CO3120 Computing with Management Project" as appropriate.
 - (b) "Dissertation" **or** "Plan" **or** "Interim Report" as appropriate.
 - (c) The full title of the project.
 - (d) The full name of the author.
 - (e) The month and year of submission.
 - (f) "School of Informatics, University of Leicester"

Please make sure this information is presented *clearly*.

4. The Plan, the Interim Report and the Dissertation must contain a table of contents. This should follow immediately after the title page. It must list, in sequence, all relevant subdivisions of the document.
5. In the Interim Report and the Dissertation the signed declaration shown below must be included appropriately before the main text begins.

DECLARATION

All sentences or passages quoted in this report, or computer code of any form whatsoever used and/or submitted at any stages, which are taken from other people's work have been specifically acknowledged by clear citation of the source, specifying author, work, date and page(s).

Any part of my own written work, or software coding, which is substantially based upon other people's work, is duly accompanied by clear citation of the source, specifying author, work, date and page(s).

I understand that failure to do this amounts to plagiarism and will be considered grounds for failure in this module and the degree examination as a whole.

Name:

Signed:

Date:

6. The Dissertation should include an abstract, which summarizes the report in about 300 words. The abstract should follow the declaration on plagiarism and before the table of contents. The abstract must provide a synopsis of the report and must state clearly the nature and scope of the work undertaken and of the contribution (if any) made to the knowledge of the subject treated.
7. The Plan, the Interim Report, and the Dissertation should contain a bibliography. This will occur at the end of the main body of text, and will be arranged either alphabetically by authors or in the order in which the references are first cited in the report. Every reference in the bibliography should enable the reader to identify the work cited and locate the specific passage referred to. For example, you might have the following listed references:

[1] Silberschatz, A. and Galvin, P.B., *Operating System Concepts*, Addison-Wesley, Reading, Massachusetts, 1994.

[2] <http://www.cs.le.ac.uk> Homepage of the School of Informatics Website, University of Leicester. September 2019.

and this would be cited in the text as [1]. An ideal bibliography will also give edition numbers, ISBN, and perhaps other useful information. For further guidance on this matter you are strongly advised to consult your supervisor.

Please note also that any quotation from a publicly available web page counts as a reference citation and must be handled as such. Failure to do so is a form of plagiarism.

8. The Interim Report may contain appendices, and the Dissertation will often contain appendices. Any appendices should follow the list of references. The purpose of appendices is to contain technical material which would interrupt the flow of the main text. Note that full program listings should not occur in the main text.

It is important for you to realize that your documents are partly assessed on the standard of presentation, including correct and clear use of the English language. Unless you are exceptionally

confident in your ability as a speller and proof reader, you should take advantage of automatic spell checking facilities available on the computer system you are using, and you should strive to ensure that your documents are written in lucid and grammatical English. If you have problems with English language composition, you should ask your supervisor for advice on material to give you help in this area.

The Teaching and Learning Unit has prepared written material on writing skills—please make use of it! They also offer tutoring on the English Language.

Remember that competence at expressing oneself clearly is a vital skill in professional life!

4 Module Organisation

You should now understand that to complete this module you need to produce the required deliverables. In order to facilitate this, you will have a supervisor that you meet with regularly, and you will attend support lectures and laboratories. These organisational matters, and other things, are explained in this section.

4.1 Lectures and Laboratories

A complete list of lectures and laboratories can be found at the “Learning Materials” section on the Blackboard.

Note, it is compulsory that you attend the scheduled lectures and Labs. The timetable for these sessions is provisional. There may be changes regarding the time and place. If so, it will be announced via the Blackboard.

4.2 Administration and Communication

You may receive broadcast emails concerning project organization and deadlines. *Please keep enough free space in your mailbox, and check your email regularly, at least every couple of days when deadlines are approaching.* If you do not receive any such emails, it may be the case that you are not currently in the mailing list. Please ask the module convener to include you.

Note, however, that assessment deadlines are given in this Study Guide, and it is your responsibility to observe them.

4.3 Allocation of Supervisor, Topic, and Second Marker

Allocation of supervisors to students was announced before the beginning of Semester 1 of the current academic year on the basis of the preferences expressed by students. Note that, in order to equalize the load amongst staff, you may not have the topic of your first choice.

The second marker of the project was announced along with your project supervisor. The second marker is actively involved in monitoring the student's progress throughout the progress of the project, and assists the supervisor in assessment, feedback and advice.

4.4 Role of Supervisor

The supervisor will meet with you fortnightly for up to half an hour in Semester 1 and Semester 2 to discuss the progress of the project. You should contact your supervisor in the first week of Semester 1 to make appointments for your meetings. The primary function of the supervisor is to *advise* you on the general aims and objectives of the project, the working methods to be used throughout the year, and the preparation of various deliverables. It is not the function of the supervisor to give instructions to you as to how to proceed, still less to do any of the work. *The project is required to be entirely your own work, and it is your responsibility to ensure its quality.*

4.5 The Student Diary

Marks will be awarded for student effort, organization, and participation. The evidence on which this is *partly based* is on each diary sheet detailing your weekly activities. A diary sheet *must be completed every week*. The supervisor may ask to read it during the supervision time. You should upload a diary each week onto SVN in the directory /docs/0_diary. This will be marked as part of the Effort and Organisation component. You can find the Diary template on the Blackboard for this module.

Each week you should record in summary the work that you have undertaken, and the work you plan to do during the coming week. If any of this differs significantly from your plan, you should say why this is so.

4.6 Progression Monitoring using SVN Version Control

All deliverables must be submitted on the school's version control system - Subversion (SVN). This is available in both Linux and Windows.

One of the main benefits of using this system is that you can revert back to previous versions. It will also give a good log of what stages have been carried out in completion of your work. This will help your supervisor in assessing how well you are getting on. **The use of SVN is a direct part of this module. Failure to show sufficient use of this system will mean that you fail this module.** This means that you will need to have your files under SVN from the start, you cannot suddenly add a large number of files that are critical to your project just before you see your supervisor or before you give a demonstration of your system. Your supervisor will need to see that you have used the system from the beginning and the different stages of committing work have been carried out. You will need to use the system in a manner that shows clearly the major changes that have been made to files by submitting modifications to SVN.

Details about how and when you have used the system are available to your supervisor, as they will have access to your repository. They will need to access your SVN repository every fortnight during the course to see that you are continually using this system and the progress being made. This means that you will need to **submit changes and newly created files at the minimum of once every two weeks**. Dates to do this will be agreed by your supervisor. If you do not regularly use SVN then an **Official Departmental Warning Letter** will be issued and copies sent to your supervisor and personal tutor. Any **further non use will result in penalties** that are at the discretion of the School's Academic Committee but total non use of the system will mean failure of this module.

To make sure again that you are using this system **any demonstration of your system** will use

a **checked out version of your repository**. Your supervisor can check then that all the files have been submitted to SVN and the date this was done last.

SVN allows you to work on a copy whilst another copy that is known to work is being used. You are able to work on another copy anywhere you have internet access. Changes then will be submitted and these will be reflected in the other copy that is being used by a simple command. There is also a way of working on a copy that you have checked out onto external disk drive and then committing changes when you have an internet connection.

To get you started with SVN there will be a **laboratory session that is compulsory for all of you to attend**. This will be in the first week of term. It is important that you attend this lab as it will show you how to setup up your repository and the basic commands associated with daily use.

5 An In-Depth Overview of The Computer Science Project

5.1 General Points

This module is assessed entirely by coursework, and consists of independent work undertaken by you, with guidance from a member of staff. The assessment is based on the quality of the various deliverables produced during the year. If your computer science project is *entrepreneurial*, you should refer to the specific study guide for the entrepreneurial projects, which can be found from the project module's Blackboard site.

Whether you are writing up an account of computing theory, or actually building a large scale software system, you should not produce deliverables *solely* by assembling together "text" or "software components" produced by other people (see Section 9 on plagiarism). Your work must contain some simple originality, and be written in your own words. Of course, some "quoted text" may be used, provided the source is clearly cited, but writing in your own words which are *clearly very distinct* from the source will obtain greater credit. Likewise "(software) components" can be used as *parts* of a project, but they should be well documented by the student, and relevant citations indicating the origin of the code *must* be made. A project must consist of more than a collection of well known, well documented "software components". A good project will deliver well engineered software that can be understood, used, maintained and evolved by people other than its developer.

It follows from this that projects should contain some work which is reasonably original and challenging. This does not mean that the work must be publishable! But it should be clear where students have delivered their own ideas, have solved practical problems (usually through the process of software engineering) and undertaken work which is clearly of a standard above that of repeating techniques learned from level two (CO2***) modules. A good project gives you the opportunity to demonstrate a wide range of computer science skills, produce novel ideas, and deliver well engineered software.

Thus projects should not consist in entirety of (potentially) very straightforward activities, involving a simple design and coding. An example would be a "web pages and a database e-commerce" project, where the design and subsequent report contains little in the way of theoretical foundations (for example, no database theory is provided, and no formal testing).

Projects must contain a detailed list of references (bibliography), and these must be cited in the text of the project, in written reports and in source code as appropriate, making it clear on what

sources of information the student's work is based.

5.2 Specific Points for Regular Computer Science Project

In undertaking your project, give consideration to the modules that are specific to your degree. In Computer Science you have taken Functional Programming, Logic and Problem Solving, and Automata Theory. Of course your project may well concern very different topics, but you are encouraged to think about how you might apply your specialist knowledge: for example if your project involves some AI then consider the role of Prolog; if you need to construct a simple parser, then consider doing so in a functional language—so much quicker to code than in say Java or C++.

Please bear in mind the accreditation requirements for Chartered Scientist and Chartered Engineer. These are specified in Section 5.3; read through them carefully. Try to adopt a scientific and engineering approach to your project work; to obtain good marks in the Computer Science project you should show some evidence of both approaches.

5.3 Accreditation Requirements for Computer Science

These differ according to the degree programme you are registered on. Chartered Scientist and Chartered Engineer and CITP apply to the Computer Science Project.

5.3.1 Chartered Scientist (CSci) For the Computer Science project *you should make sure that your project addresses the relevant concerns of the Science Council*. Your project should try to demonstrate that you are able to

- deal with complex scientific issues, both systematically and creatively,
- exercise self-direction and originality in solving problems, and exercise substantial personal autonomy in planning and implementing tasks at a professional level,
- make sound judgements in situations where there is any absence of complete data and be able to communicate your work clearly to specialist and non-specialist audiences, and
- show a commitment to continuing to advance your knowledge, understanding and competence to a high level through your career after your project.

The Science Council also requires that, long term, you

- demonstrate an understanding and commitment to Health and Safety and environmental issues related to employment, and
- comply with the relevant Codes of Conduct

and note that you can find out more here

<http://www.charteredscientist.org>

<http://www.charteredscientist.org/about-csci/standards>

You should keep in mind, at all stages of your project, the process of scientific thinking and problem solving. Have in mind your ideas and assumptions (hypotheses) about what you do, and draw your conclusions as you proceed. You should try to make sure that throughout your project, where applicable, you employ sound scientific and mathematical principles to underpin your work. For example, you might use CCS to model a concurrent system, algebra to verify an algorithm terminates, perform experiments to test a hypothesis you have about a dataset, calculate algorithm complexity or memory requirements, and so on.

5.3.2 Chartered Engineer (CEng) You should also employ sound principles of (computer) engineering, and these have been taught throughout your degree. You might like to view the Engineering Council web pages here <http://www.engc.org.uk/>. For example, when designing an OO software system, use a suitable design methodology such as UML; when testing software gives a sound engineering account of the types of testing available and their suitability to your project; when creating a large scale system, use the engineering techniques such as a unified approach; and so on.

5.3.3 Chartered Information Technology Professional (CITP) You can find out more about CITP here <http://www.bcs.org/category/10972>. In undertaking your project, use your knowledge and understanding of

- the methods and issues involved in deploying systems to meet business goals
- methods, techniques and tools for information modelling, management and security
- systems architecture and related technologies for developing information systems Computing-related practical abilities

wherever relevant, and

- use appropriate theoretical and practical processes to specify and deploy, verify and maintain information systems, including working with technical uncertainty
- define a problem, research its background, identify constraints, understand customer and user needs, identify and manage cost drivers, ensure fitness for purpose and manage the design process and evaluate outcomes
- apply the principles, methods and tools of systems design to develop information systems that meet business needs

5.4 Sample Illustrations of the Content of Computer Science Projects

- If you write a Theorem Prover then you should be able to demonstrate the scientific/mathematical principles on which the tool is based. Thus any implementation should be supported by a formal description of the logic, and ideally some supporting properties of the logic.
- A project on parsers should not consist of just a written account of (well known) parsing techniques, but should include some foundational material on automata or other models of

computation. If standard lexer/parser software is used by you, you should cite its use carefully: you might use it in an original way to solve a problem. Alternatively, a student might undertake to write their own parser. The key point is that for a *good mark*, *there needs to be some originality*, whether it results from working from first principles, by thoughtful application of known techniques, or by making use of novel architectures.

- Setting up a client-server program in Java/C++ is standard material. Design and implementation of such a program is well documented. So projects involving such architecture must contain design and development of further tasks which might build upon the standard material. Computer Science projects should also include some background material on the theory of client-server protocols, and make use of formal models wherever possible.
- Creating PHP web pages that implement login scripts and use a shopping basket is standard material. There are easy tutorials on the web, including simple SQL database examples. Again, projects involving such architecture must contain design and development of further tasks which might build upon the standard material. (A “Photo Album” project falls into this class of projects, and original problem solving material might consist of on the fly image compression or/and user configurable portals.)

6 Feedback

The School, in line with University policies attempts to mark assessed coursework within ten working days. However, given the nature of the assessment for this project, the turnaround time for marking some assessments may be more than ten working days. Feedback of all assessments will be given online from the Student Feedback Webpage at https://campus.cs.le.ac.uk/ProjectSystem/2019_20/undergraduate/mvc/index.php/campus/progress/progressPage

The expected feedback date for each assessment is given below:-

- **Plan and Description:** feedback will be given by **Sunday, 3rd November 2019**.
- **Software Prototype:** feedback will be given by the end of Semester 1 (**Sunday 19th January 2020**).
- **Interim Report:** feedback will be given by the end of Semester 1 (**Sunday, 19th January 2020**).
- **Interview:** feedback will be given within ten working days after the deadline of assessment.
- **Final Report, Software Product, Technical Achievement, and Mini Viva:** feedback will be given after the Exam Panel & Board in **late June**.

The following table gives another summary of all project components and their contribution towards your final mark.

Project Component	% of Final Mark
Plan	Pass or Fail
Prototype and Demonstration	3%
Interim Report	7%
Interview	5%
Dissertation	25%
Mini Viva	10%
Effort	5%
Software System	15%
Technical Achievement	30%

7 Students with Specific Learning Difficulties and Long Term Conditions

The *AccessAbility Centre* of the University offers services and support for students with dyslexia, and other specific learning difficulties, disabilities and long term conditions. Each academic School has an *AccessAbility Tutor*. In Informatics this role is performed by the Welfare Tutor, *Dr Fer-Jan de Vries*. He is the person within the School who you can talk to about any disability-related issues. You can find more details in the Student Handbook in the *Personal Support for Students* section. Assessment in Informatics can take many forms. You may have been provided with an *Assessed Work Cover Sheet* by the AccessAbility Centre which you should attach to your written assessments so that your work can be properly assessed by the module convenors. You might also require special arrangements for class tests and oral assessments (such as additional time, quiet rooms or handouts in specific formats) in which case you should discuss your needs with the AccessAbility tutor at least 7 days prior to such assessment for appropriate arrangements to be made.

8 Late Submission of Project Coursework

We adopt the standard University Rules of Assessment:

- www2.le.ac.uk/offices/sas2/regulations/documents/senatereg7-assessment.pdf
- www.le.ac.uk/sas/assessments/late-submission.

Roughly speaking, you may submit after any of the submission dates, but *you will immediately lose 10% (of the maximum mark available, even if only a minute late), and a further 5% for each further whole day late*. Thus it is still *essential that you submit your coursework/deliverables in time* in order to ensure you have obtained the best mark possible.

In the event of you being unable to do coursework because of illness or other bona fide reason, allowance will be made provided you submit a mitigating circumstance form, which is accepted. Please refer to your student handbook for more details on this.

9 Plagiarism

The issue of plagiarism is very important. You **MUST** read the University's statement and the school's regulations concerning plagiarism. These can be found in the University Regulations at <https://www2.le.ac.uk/offices/sas2/assessments/plagiarism>

and in the Student Handbook in the *Referencing and Academic Integrity* section.

The University regards plagiarism and collusion as very serious offences and so they are subject to strict penalties. The penalties that schools are authorised to apply are defined in the Regulations governing student discipline, see

<https://www2.le.ac.uk/offices/sas2/regulations/senate-regulations>