University College Dublin
An Coláiste Ollscoile, Baile Átha Cliath

MSC COMPUTER SCIENCE (CONVERSION) RESEARCH PRACTICUM COMP47360

June 2017

Guidelines

Contents

Research Practicum COMP47360

GUIDELINES

INTRODUCTION

The Research Practicum constitutes a vital component of your degree and as such requires careful consideration and effort. Every MSc Computer Science (Conversion) student is obliged to undertake a project as part of the research practicum. The purpose of the project is to provide students with an opportunity to learn how to undertake a major project, taking it from problem specification through to problem solution while working as part of a team.

The purpose of this document is to outline the nature of the project, to explain the procedures involved in undertaking such a project, and to provide guidelines on the important task of report writing. Please bear in mind that this document is intended only to describe the process and the marking scheme in an indicative way, and in appropriate circumstances certain details may be changed during the term.

The team-based project should be approached and completed in a spirit of collegiality. The purpose of the practicum is to learn, bring together your experience, and demonstrate the skills that you have acquired in the programme to date. The finished project can be used as a show-case for employers to demonstrate how you have taken a project specification and delivered a solution.

TEAMS

The project for the research practicum is a team-based one. Students will be arranged into teams of approximately four members. The allocation of students to teams is based on students expressing an interest in the specific roles which have been identified as important to the success of the project. This approach will ensure a balance of interests on teams. While individuals will take a lead role in a certain area of the project, all team members are expected to contribute to all aspects of the design and development of the project. A description of the roles and the role selection form can be found in Appendix I. Students are required to produce a team agreement in week one of the project. There is no set template for this but the team agreement should cover the ground rules for communication, participation, meetings, conflicts and decision making.

COMMUNICATION, PEOPLE AND WORKSPACE

We will communicate with you in various ways regarding project guidelines, deadlines, etc. Announcements may be made at meetings, emailed to you, or posted in the CS Moodle page for COMP 47360. Please be sure to keep yourself informed of what is going on by looking at the schedule. The people involved in the module are listed below. The module coordinators should be the first point of contact for the module.

Project Mentors

Name	Contact
Gavin McArdle (Module Coordinator)	gavin.mcardle@ucd.ie
Vivek Nallur (Module Coordinator)	vivek.nallur@ucd.ie
Julie Berndsen	julie.berndsen@ucd.ie
Pádraig Cunningham	padraig.cunningham@ucd.ie

Teaching Assistants

Name	Contact
Hamed Jahromi	hamed.jahromi@ucdconnect.ie
Ellen Rushe	ellen.rushe@ucdconnect.ie

Career & Skills Consultant

Name	Contact
Edel Caraway	edel.caraway@ucd.ie

Project Mentors

The project mentors play an important role in guiding and advising you as you work on your project. You should be sure to make good use of the mentors, and to let them see that your approach to your project work is organised and competent. This can be demonstrated in the informal meetings in the workspace, during presentations or at team meetings with mentors. In all cases, be professional: if you cannot make a meeting be sure to inform them in plenty of time. Do not come to the mentors with every little problem you have, try to work things out yourself as a team. This will allow you to have an informed discussion with mentors about the problem. At the same time, if you are having serious problems with your project and progress is halted, be sure to let the mentors know as soon as possible.

You will hopefully come to regard the mentors as friendly and supportive. If you have any problems during the term that are impinging on your work, let us know. In circumstances such as illness or bereavement, we are flexible in the allowances we can make.

Teaching Assistants

The teaching assistants are there to support you through the project process. The teaching assistants can provide technical help and guidance on the project process and progress. They cannot do the work for you but may be able to help you find a solution to specific issues you are having.

Career & Skills Consultant

The career and skills consultant has put together a programme of career planning talks and speakers from industry. These lunchtime sessions occur on Wednesdays as per the schedule on Moodle. It is expected that all students will attend these sessions.

Workspace

Classrooms B1.08 and B1.09 in the Computer Science Building are available for use during the summer term should you need them. Additionally, there are meeting locations available in the science building if you wish to use those. Other rooms in the Computer Science Building may be used if they are vacant.

Meetings

Each team member will deliver an individual presentation which represents the work of the team on a topic (See Section entitled Project Presentations). This is an opportunity to receive feedback from the project mentors directly but also classmates and other teams. The project mentors will call to the workspaces on Wednesdays (excl. Presentation Wednesdays) to receive updates from teams so it is expected that team members will be present. Additionally, the teaching assistants will also be in the workspaces on Wednesdays to assist and provide guidance on the progress of your team. If you wish to have a separate meeting with a teaching assistant, you may also make an appointment with them. You can make an appointment with a project mentors or teaching assistants on Wednesdays to discuss your project or issues and challenges you are facing.

ALLOCATION OF MARKS

Your project grade comprises a large proportion of your final award in Computer Science. The breakdown of the grade is as follows:

Component	Weighting
Individual Presentation	5%
Group Presentation	5%
Group Portfolio	20%
Individual Report	70%

The grades for your Group Portfolio, Individual Report and Presentations are awarded by members of the academic team. Note that at each component is graded – using letter grades and not a numeric score.

PROJECT PRESENTATIONS

Each team member will deliver an individual presentation which represents the work of the team on a topic. These topics and their timings are specified in the table below. Additionally, there is a final presentation to showcase the work produced by the team. The presentations take place before an audience that will normally include project mentors, your team and classmates.

Date	Presentation Title
14-June-2017	Initial Big Picture
28-June-2017	Detailed Requirements and Architecture
12-July-2017	Prototype and Data Analytics
9-August-2017	Pre-final Prototype
23-August-2017	Final Presentation

Presentations and possibly a demonstration (depending on stage of project) will typically be 10 minutes, with a further 5 minutes for questions. Please note that all students are obliged to attend the presentations.

What makes for a good presentation? Firstly, to present your teamwork well, you must take ownership of it but you need input from your team. This means that you describe the work, and the motivations. If you do not communicate your own interest in your project, no one else will be interested either. Explain why you did things a certain way, explain the process, and be sure to motivate the problem for a general audience. Always build your presentation around a strong working example or use-case. Do not use overcomplicated examples that require too much prior understanding and too much explanation during your talk.

Your presentation should have a narrative structure. It should tell a story with a beginning, a middle and an end. A good presenter is a good story-teller. Set the scene well, motivate your problem, and give your audience enough information to anticipate your next steps throughout. Work on the ending, it is what people will remember most. Story-telling takes practice. It is not about how much you can fit onto your slides, but how you can help your audience to remember. This is definitely a skill worth honing for your future career.

Project Presentations Assessment

Marks will be awarded for your presentation by the project mentors based on the following criteria:

- **Technical Merit:** On a technical level, how good is the work that is being presented?
- **Organisation and Delivery:** Is the presentation well-structured and delivered clearly? There should be a logical organisation to the presentation and it should be easy to follow.
- **Handling of Questions:** You should be able to field questions on any part of your project work.

A little note on answering questions: It is vital that you listen carefully to each question and be sure that you understand it before attempting to answer. Query the questioner if you are not sure what their question is. If you feel the questioning is tangential to your project work, do not be afraid to say something like "That's an interesting question, but my project is focused on the area of…". Do not be afraid of tough questions, or of giving tough answers. Tough questions allow you to shine, and to give the graders one more reason to elevate your project.

PROJECT REPORTS

There are two reports which you must write as part of your project. The Portfolio, which is a group contribution is due at the end of July, and the Individual Final Report is due at the end of the project. All reports should be produced in a professional manner. The Section entitled 'Preparing your Project Report' provides detailed guidelines and advice on the writing of the final report, much of which applies to the other report as well. You will be provided with a LaTeX template that defines the formatting and style of your reports. This saves you a lot of work, and introduces a standard layout to all the reports. These template documents are to be used in writing all your group and individual reports.

GROUP PORTFOLIO

Each team should produce one group portfolio which provides a succinct description of the application which your team are developing. It should describe how you are addressing the project specification and the progress in development.

The format of the portfolio report is fixed and its length limited to five content pages. Apart from a cover page, a problem description, table of contents and references, there will be three content sections as follows:

- Introduction (suggested 0.5 pages): This describes the content of the remainder of the report.
- Description of Solution/Application (suggested 4 pages): This should describe the approach you are using to address the problem and produce a prototype.
- Progress Report (suggested 0.5 pages): This should describe where you are in the development process and what remains to be completed (with a timelines).

Please observe these rules carefully. They are not just optional guidelines!

Group Portfolio Assessment

Marks for the Group Portfolio will be awarded according to:

- Grasp of the problem
- Quality and extent of proposed solution
- Progress achieved to date and future schedule

FINAL REPORT

Each student is required to produce an individual final report which describes the project from their point of view. The report should describe the research which you carried out, your role in the design, development and testing elements of the project and any innovations that you were responsible for.

The format of the final report is not fixed in the way the portfolio report is, but its length is strictly limited. Apart from a cover page, project specification, abstract, table of contents and references, the maximum number of content pages is 25. You do not need to rewrite the *description of the solution* from the Group Portfolio but can include this as a section in the final report to make it complete. These page limits mean that you do not need to "pad out" your report. At the same time, it does not make your task easy. Recall the famous quote: "I am sorry for the length of my letter, but I had not the time to write a short one."[1] The Section entitled 'Preparing your Project Report' provides detailed guidelines and advice on the writing of your final report.

Final Report Assessment

The project report is a very important part of your project and its preparation and presentation should be of extremely high quality. Remember that a large portion of the marks for your project are awarded for this report. In the assessment of a project, the aspects that may be taken into account include the following:

- Problem Comprehension: Does the student clearly grasp the problem that they are trying to solve?
- Technical Content and Overall Approach: Is the technical work of a high standard? Has the student used appropriate techniques in their work? Is the design compact and elegant, or is there clearly a better approach that could have been used?
- Justification: Does the student justify their approach? Where there are a number of options, they should be enumerated, analysed and the most appropriate one chosen.
- Critical: Good science is always self-critical, and this often does not come naturally. A good project report will, in its conclusions, try to be critical of the work done, highlight its weaknesses and failings. This is not a sign of a poor project; quite the contrary. A project report that praises the work done uncritically is lacking in an important respect.
- Completeness: Is the project work complete? The project description makes it clear what should be achieved so the question is to what extent has the student succeeded? A project that does not achieve all its aims but provides a clear justification of why this happened is also a good result.
- Report Quality: Is the report well-written and presented? There should be a clear structure to the overall document and a compelling "storyline" running through each chapter. Needless to say, bad spelling, poor punctuation and ill-formed sentences make a report hard to read and will result in a loss of marks.

[1] http://quoteinvestigator.com/2012/04/28/shorter-letter/

- Outcome of Project: Was the project (as specified) successful? Did the project go beyond the specification in an innovative way? E.g. if software was produced, is the code correct? It is expected that demonstrations of your application have been made. An implementation that is hard-to-use or buggy will not score well. The source code is also part of your final submission and this should be well-designed and laid out.

Don't get depressed looking at this list! It is important that you understand what a good report is and show that you are striving towards this ideal. There is an excellent textbook by Christian Dawson on this topic that the School strongly recommends[1]. It is available from the campus bookshop or from Amazon.

Any implementation produced should also be submitted. Not all implementations will be scrutinised, but the factors that may be taken into account in assessing them include:

- Correctness: The software should perform as described in the final report. Minor deviations are not a problem, but any major aspect described in the report should also be present in the software.
- Robustness: The software should operate reliably and be not subject to frequent abnormal terminations.
- Code Quality: In your years in UCD you have encountered many design and implementation heuristics. It should be apparent in the code you have written that these have guided your work.

Needless to say, a violation under point (1), that is to say a flagrant misstatement in your report of what your implementation actually does, is a very serious matter.

SUBMISSIONS

Project Report Submission

Submission of all reports is via Moodle. You will be provided with details of this later in the term. Late submission will result in a sanction except where documented extenuating circumstances exist. Late receipt of a project report will normally be penalised by a loss of a full 5% for each day, or part thereof. This means that a portfolio that is deserving of a final mark of 65% will actually be awarded only 60%, if Group Portfolio is submitted a single day late (weekends included). Please understand from this that it is simply not acceptable to be late in the submission of any project report. A softcopy in Portable Document Format (PDF) must be submitted. No hardcopy submission is required.

Project Code Submission

The project involves the development of a piece of software or application. This must also be submitted, along with information on the development environment used and instructions on executing the application. The exact details regarding how to submit your code will be provided, but the deadline will be close to the final report deadline.

Plagiarism

Plagiarism is using other people's work in your project without acknowledging the fact that you are using it. It is most likely to occur in writing the Background chapter, as this will be based on material that you have read.

First of all, here is a tip for avoiding plagiarism. When you are actually writing a chapter, have none of the sources you are using open in front of you. Read the source material first, assimilate it, and write the description in your own words. In this way, the text you write will really be your own understanding of the area. Do provide references to your source material as well.

The School deals very firmly with instances of plagiarism, so please be very careful in this regard. It is also very naive to plagiarise in a field when your supervisor is likely to be very familiar with the source material themselves.

Partly why this is so important is that most students who plagiarise do so without realising what a serious issue it is, or what the consequences will be if it is discovered. Please be extremely wary of how you treat source material in your reports. There are a number of tools available on the web page for the projects to check your content yourself before submission. Ignorance of the rules is never a good defense. You should familiarize yourself with the UCD School of Computer Science Plagiarism Policy which is available on Moodle.

Backups

This is so important that it cannot be overemphasized. Keep several backups of your work in various locations, so in the worst, worst case you still have something to roll back to and are not left with nothing.

To assist you with the management of your project source code and reports, a git repository management system is maintained at https://csigit.ucd.ie. You may use this system creatively to backup, log and share weekly progress reports and project outcomes. It is strongly recommended that you keep your project directory synchronised with your git account, with weekly commits reflecting your progress as a minimum interaction level with the system. Your git repository should be suitably structured e.g. weekly reports, research documents, project implementation source code, outcomes/results. A snapshot of your git repositories for the past week will be kept, should you need to recover lost work.

PREPARING YOUR PROJECT REPORT

Report writing is an important skill. No matter what field you are engaged in, you will almost certainly find it necessary to be able to write a clear report on your work. If you have a talent for technical writing, you will no doubt find it an easier task. However, it is a skill that can be acquired with practice and it is an essential part of your project work. Be sure to allow yourself enough time to write the report; the process generally takes at least two weeks.

Below, an approach to take in structuring and writing your report is suggested. It is not carved in stone; feel free to adjust this to suit your own particular project/style.

Create a Report Structure

The first step is to produce a draft table of contents, showing how the entire report is to be structured into chapters, sections, and even subsections. Annotate each item with the purpose it is to serve in the overall report, and its anticipated length in pages. When you have done this ask yourself the following questions:

- Is there a logical flow through my report? If it does not flow logically at this high-level stage, it certainly will not flow well in the end either. Move sections around until you feel there is a logical thread running through the document.
- Have I written about all the important issues? Pull yourself back from the report and think about the project in general. You should not write about everything you did. This is a report, not a diary but do not omit any vital sections either.
- Are the issues that I have written about important? You have probably written sections that should really be omitted. It is tough to cut out a section that you have laboured over, but dross in a report has a very negative effect on overall quality.

Once you have a sound overall structure, you can start writing sections knowing that they fit into an overall plan. You will know how much preparatory material will have preceded each section, and you will know to what extent it is expected to lay the groundwork for later sections. You may find that you have to change the structure later in the writing of the report. As with software, the later you change the design, the more work it entails.

How Many Chapters?

The details of the structure will of course depend on the content and nature of the project you are working on. Generally, you should break down the report into approximately six chapters. One possible template you could use is detailed below, but remember that this template is only for guidance. You may decide to merge some chapters, or have an extra core chapter. It all depends on your project and how you wish to present it.

Title Page, Project Specification, Acknowledgments and Table of Contents

The title page should state at least the project title and your name. As with the Interim Report the full project specification should be reproduced here. In your Acknowledgments section, give credit to all the people who helped you in your project. A Table of Contents is essential. The order of these is usually Title page, Project Specification, Abstract, Acknowledgments and Table of Contents.

The Abstract

The abstract should provide a short overview of your project that enables a reader to decide if your report is of interest to them or not. It should be concise, to-the-point and interesting. Avoid making it read like a verbose table of contents! Avoid references, jargon or acronyms, as the reader may not be familiar with them. An abstract usually contains a brief description of:

- the project and its context;
- how the project work was carried out;
- The major findings or results.

One paragraph is plenty. The main thing to remember is the principle that the abstract must be short, and a person reading it should be able to determine if they want to read more. For example, if your project involves building a compiler for Java, and a major section of your work is focussed on developing an efficient parser (rather than say code generation), make this clear in the abstract. Then a reader who is interested in efficient parsing techniques knows that your report may be of interest to them.

Chapter 1: Introduction

Some topics it may contain include:

- A discussion of the original aims of the project, and the modified aims if appropriate and any innovations;
- The scope of the project and a general justification for the work undertaken, perhaps providing a brief background description;
- A description of the structure of the report, i.e., a road map for the reader.

Chapter 2: Background Research

The contents of this chapter depend on the nature of your contribution to the project. If you are working on a research-oriented project, then you will present the research landscape within which your project is being conducted and consider approaches that have been adopted by other researchers. In a development project, you may describe the domain in which you are working and the technologies and programming tools you are using. Tutorial-type descriptions are never appropriate.

Chapters 3 and 4: The Core Chapters

These are the principal chapters of your report and their structure will vary from project to project. The aspects of your project that you will describe in these chapters include:

- A detailed account of how you approached your project, i.e. the strategy you employed. This should be at a high level, separate from design and implementation issues.
- A discussion of the design aspects of your project. Include here a discussion of interesting problems you encountered and the alternative solutions you considered.
- Innovations that you have incorporated.

Use the appropriate notations and formalisms in this chapter. Everything you have studied in your degree is relevant here. If there is a crucial algorithm at the centre of your project and its performance is important, attempt to provide an analysis of its complexity. If you have performed an object-oriented design, use the

appropriate UML models to describe your work. Do not mindlessly produce "documentation", but think about what you want to communicate to the reader and use the most appropriate method of doing so.

Chapter 5: Detailed Design and Implementation

In doing your project work, a lot of time will be spent on detailed design and implementation. The nature of programming is that it is a very time-consuming task, and even for experts a "silly" runtime error may take days to correct. In spite of this, this chapter should not be the main focus of your report. Make it clear what implementation technology you used and discuss any interesting implementation issues that arose. For example, if you were using a particular data structure that had to be optimised in a certain way to be suitable for your project, describe it in this chapter. On the other hand, if you used an obvious/standard approach, then there is no need to devote much space to it.

Chapter 6: Testing and Evaluation

You may decide to merge this chapter with another, but I have described it as a separate chapter as it is very important in its own right.

If the focus of your project is the development of a piece of software, then you should address the issue of how you demonstrate it to be correct. You will not have time to really test your software in the way that industrial software is tested. However it is important to show that you have taken a methodical approach to testing and that you have tested your software in such a way that you are justified in having some confidence that it is correct.

Another type of project involves designing a heuristic or approximate solution to a challenging or ill-defined problem, e.g., to develop a junk mail filter or to mine a certain type of data from the web. In this case the precise desired behaviour of the software is hard to specify (what is junk mail anyway?), so it is more appropriate to describe how you evaluated the solution. This will involve running a number of experiments and presenting the results. As with testing, this is a complex area that you should spend some time coming to terms with.

Conclusions and Future Work

If you are writing this chapter bleary-eyed and caffeined-up on the day of submission, you are not going to do your project justice. This is a vital chapter in the assessment of your work. Academics, in getting the feeling for any type of report, will typically read the introduction and conclusions first. Your conclusions should not read like "I did all this stuff, it went great, and here's other stuff someone else might do." This chapter should cover the following areas:

- Conclusions: In a research-oriented project you will state the overall conclusions you have come to. In a development project there may not be a conclusion as such, so just state what has been achieved. Be very critical in this section. Describe the weaknesses of your approach and avoid making unwarranted conclusions.
- Future Work: Think carefully how your work might be extended or applied to another domain. There will probably be some obvious extensions. If you are able to propose some interesting ideas

that are not immediately apparent, this demonstrates that you have a clear understanding of the field.

It is good scientific style to make strong statements. If a certain statement is warranted by the results of your project, don't be afraid to make it. Strange though it may seem, a strong statement that turns out to be wrong is better than one that is vague and wishy-washy. The former can lead to a lively debate where the truth may emerge, but the latter will produce meaningless agreement, because it ultimately says nothing.

References

Use one consistent system for citing works in the body of your report. Several such systems are in common use in textbooks and in conference and journal papers. Ensure that any works you cite are listed in the references section, and vice versa. LaTeX will manage the referencing for you, and be sure to make use of this facility. It may take more time in the beginning, but at the end of the write-up it will certainly have saved you a lot of time.

In approximate decreasing order of quality, the best sources to cite are journal papers, international conference papers, national conference papers, books and web pages. Don't just supply a URL if there is an equivalent conference paper you could cite instead. Also, it strengthens your project if at least some of your references are to recently published material.

Appendices

Material that you want to include in your report, but that is not directly relevant to the main thread of your report, can be put in an appendix. Possible examples include program/code listings, detailed test results, user guides etc. In most cases, appendices are not necessary and it is only in an exceptional case that it is useful to provide a code listing. Remember that material in the appendix counts towards report length, so do not exceed the limits defined in sections 5 and 6.

Order of Writing

The previous section dealt with one possible logical structure for your report. The order in which you write it all is another issue. There are no fixed rules here. Some people like to write notes throughout the project, so when it comes to writing the final report, they already have a lot of material prepared. This is a very valid idea, but avoid wasting time writing very polished notes during your project work. The notes/sections you write can be quite rough, and only in the final report do you bring them up to full report quality. The reason for this is that you may have to tailor them considerably to fit the context of the report, and this will mean that much of the polishing will have gone to waste.

Assuming you have created a report structure, a good way to continue is to write the introduction in draft form. The reason why you write this in draft form is that you are not yet sure what you are introducing! Only when the later chapters are completed can you return and finish the introduction.

Next the Background chapter can be included in the final report. Again, you may find that when you write the core chapters later, some of the background work becomes irrelevant and can be removed. This may seem like wasted effort, but if it results in a tighter Background chapter, do it.

Next are the Core chapters, followed by the design and testing chapters. When these are complete, you are in a position to write your Conclusions chapter, and to return to the Introduction and Background chapters and bring them to completion. Finally, write the abstract.

The next step depends on how much time you have left. Ideally you will reach this point where you have a first full draft with at least a week to go. Proofread the report once yourself, and pass it on to other people who can read (part of) it and give you any sort of feedback. Take a rest yourself, so you can return to it in a day or two and re-read the report with a fresh mind.

Note that at this late stage you can only make local improvements to the report. It is too late for major overhauls, so at this point the importance of creating a good overall structure becomes clear. If you have started with a good structure, you can aim to create an excellent finished product. However if your initial structure was awkward, the final report will not read well no matter how you tweak it.

Other Advice

This section contains a number of guidelines that are worth bearing in mind when writing.

Continuity

You may not realise this, but a good academic paper or report, like any good novel you have read, tells a story. It is valuable to keep this in mind when you are writing your report. There should be a storyline running through your report and you should make it easy for the reader to hang on to this storyline:

- At the end of the introduction provide a short description of the layout of the remainder of the report.
- Start every chapter with a brief recapitulation of the story so far, and an overview of what the chapter is going to add.
- Finish every chapter with a summary of the material in that chapter, and state how it relates to what follows.

In the core chapters, you should take care to make absolutely clear the logical connection between the overall project design and the detailed problems you discuss. If the reader is mired in a detailed description of your solution to some intricate problem, they will be encouraged to persevere if you have clearly indicated its place in the overall project.

This continuity material may sound unnecessary and redundant, but it is useful to the reader. It may help for you to imagine that the reader is coming back to your report after a break of a few days: they will be greatly assisted by occasional reminders of what has already been said.

Presentation Issues

Focus on expressing your ideas clearly. Part of your report is of course its physical layout and use of diagrams. Try not to put too much time into this. Simple diagrams are fine, and avoid the use of colour unless it really contributes something in particular. Do not bother with tricks like adjusting spacing or margins or fonts in order to make your report seem bigger or smaller. Do include screen shots of the applications and its features.

Textual Matters

Who do you have in mind as you write your report? A good model to use is that you are writing to an educated computer scientist who is not directly involved in the field of your project. Your report is intended to be a technical document, so follow the style you see in the best scientific literature that you read.

Keep your language clear and avoid colloquialisms and abbreviations. Avoid writing in overly "academic" tones. In good academic papers you will find a simple, clear style that is easy-to-read and not overwrought. The previous sentence could also be written as: "You will find, in academic papers of good quality, a style that is at once both clear and indeed easy-to-read, without being in any sense overwrought." This style is only suitable if you are making a crowning point that you wish to emphasize heavily.

Do not use the word "I". If at some point you really wish to express a personal opinion, use a phrase like "it is the opinion of this author that. . . ". Avoid over-using "we" as well, but don't use the passive voice all the time, or your report will be unreadable.

Avoid overuse of italicisation, underlining, bolding, or other devices for emphasis. Underlining is best avoided, as this was originally a device for informing a typesetter to use italics, and not a form of highlighting in its own right at all.

Do not place large blocks of code in your report. If you are considering putting code in your report at all, ask yourself first if an algorithm written in pseudocode is more appropriate. Any code you do present should earn its place and should be impeccable in construction and layout. Use a suitable font for code, such as courier, and stick to this consistently.

Pay attention to spelling, punctuation and sentence structure. Poor spelling can be very intrusive and is unforgivable given that your word-processing software surely provides a spelling checker. Poor punctuation can destroy the meaning of a sentence. If you are not sure how to use punctuation, use fewer commas rather than more. Long sentences that are difficult to write are usually also difficult to read, and may turn out in fact not to be sentences at all. If a sentence feels unwieldy, split it in two.

REFERENCES

[1] Christian Dawson, The Essence of Computing Projects – A Student's Guide, Pearson Education, 2000

APPENDIX I

For this practicum, all team members will participate in research and development but individual team members will take a lead role in some aspects of the project. These roles in an AGILE Project are listed below.

Role	Possible Responsibilities
Coordination Lead	Coordinates and solves group problems, leads and guides development sessions.
	Measures the group progress by measures defined by the team and the customer; manages the team diary.
	Makes sure that the team works according to the defined development process, answers questions/finds answers related to the methodology, looks for solutions to problems.
Customer Lead	Performs an ongoing user evaluation of the product (collects and processes feedback received from users), serves as the user interface designer.
	Tells customer stories, makes decisions pertaining to each iteration, provides feedback, and defines acceptance tests (are project specifications met?).
	Defines (with the customer) and develops acceptance tests, inspires a test-driven development process.
Code Lead	Maintains current design, works to simplify design, searches for refactoring tasks.
	Establishes testing guides and supports others in the development of tests.
	Establishes the integration environment; encourages rules pertaining to the addition of new code.
Maintenance Lead	Plans and organizes iteration/release presentations, demos and posters.
	Plans and organizes the project documentation: process documentation, user's guide, and installation instructions
	Maintains source control, establishes and refines coding standards and guides.

Examining the table above, please rank your preferences for each of the roles within an AGILE project. *

	1	2	3	4
Coordination Lead	○	○	○	○
Customer Lead	○	○	○	○
Code Lead	○	○	○	○
Maintenance Lead	○	○	○	○

Figure 1. Team Role Questionnaire