

On Writing a Final Year Project

John Smith

Supervisor(s): Johnny Smithy and Jon Smythe

0.8UOMcrest.pdf

Faculty of ICT
University of Malta

May 2014

*Submitted in partial fulfillment of the requirements for the degree of B.Sc. Computing
Science (Hons.)*

Abstract:

Here is a one page summary of the project. The emphasis should be on motivation, challenge, solution proposed and results. It should give the reader an opportunity to assess in a few seconds whether it is of interest to him or her.

It is worthwhile remembering that this is not a murder mystery, so tell the reader what you have achieved without forcing him or her to read through the rest of the report before they can understand the results of the project.

This abstract should not exceed one page, and does not count towards the page limit.

Contents

1	Introduction	1
2	Structure Guidelines	1
3	Style Guidelines	3
4	Conclusions	4

On Writing a Final Year Project*

John Smith[†]

Supervised by: Johnny Smithy and Jon Smythe

May 2014

Abstract: This is a short abstract of the final year project, a summary of the longer one given in the second page of the report. Carefully read through the abstracts of a number of papers to understand the tone and manner in which this part should be written.

This document acts as both a style guide for students and as a template which can be edited as a starting point for their report.

1 Introduction

Please allow me to introduce myself. I am Smith, John Smith. I have worked on my final year project until it drove me crazy, driving my supervisors crazy in the process. This report is the fruit of that craziness, which I hope I can share with you. Luckily I had Muxxu my hamster to keep me company. I hope you have a soulmate too.

Note: John is indeed crazy. Do not introduce yourself here, but motivate your work, introduce your project results, and present an outline of the document. Study the introduction of a good paper or two and emulate their style.

2 Structure Guidelines

Think of your report as the telling of a story. There is no fixed structure your report has to adhere to, but there are some parts of the story which will invariably manifest themselves in the report. Below is a list of typical parts which may or may not be sections in the report:

*The research work disclosed in this publication is partially funded by the ABCDEF Scholarship Scheme. Include this only if you are required to acknowledge funding or support, and not to acknowledge the support of Muxxu the hamster.

[†]Submitted in partial fulfillment of the requirements for the degree of B.Sc. Computing Science (Hons.).

Abstract: The abstract should act as a stand-alone (very) brief description of the whole story: The context, the solution, how effective it was found to be. There is no better way to learn how to write an abstract than by carefully reading the abstracts of good papers. This is usually the last part of the report to be written.

Introduction: This is one of the few parts in this list which you can never do without as a separate section. Motivate the problem which your research has been addressing, and introduce the problem, your solution and informally present the results. At the end of the section, one usually explains the structure of the rest of the document.

Background: For the reader to understand your work, he or she will have to understand a number of technical techniques or results. This is where they go. For instance, if your work is about improving on a particular compiler optimisation X using a static analysis technique Y , in this section you should explain X and Y in enough technical detail for the reader to be able to understand your results. Do not include unnecessary things just to show that you read a paper here. Using the same example, you should not explain other static analysis or compiler optimisation techniques not relevant to your approach.

Your work: Finally comes the part where you explain your results. It is crucial that where different design or solution options were available, you discuss them and why you went for that particular solution in this section. Remember also that your report is not system documentation — you need not give details about any programs you have written (unless they are directly relevant to your project). Using the compiler optimisation example, in this part of the write-up you might want to present (i) how your solution works, the algorithm using input from the static analysis and changing the code for greater efficiency; (ii) discuss alternative solutions similar to the one you propose, using the same information; (iii) explain how the implementation requires the compiler to keep an additional lookup table, and how it impacts the compiler's performance.

Evaluation: A final year project is not a programming challenge, but a scientific endeavour. It is crucial that you evaluate your results, whether it is a qualitative discussion, an empirical study or a formal analysis. Going back to the compiler example, you might give a complexity analysis of your optimisation algorithm and a set of experiments showing the gains using your approach against no optimisation and against

Table 1: Note that table captions go above the table

Name	Age	Smithsonian factor
John	21	1.2
Johnny	53	1.8
Jon	35	1.3

a related optimisation technique proposed in the literature on a number of different compiled programs. Remember to explain what question you set out to answer before you explain the experiment or analysis.

Related work: Now is the time to compare your work with that of others. While the background gave the reader the tools to understand your work, this section explains results from other papers, relating them together and with your work. There is nothing more frustrating than finding a list of paragraphs each describing a different paper without synthesising how the techniques are linked, and how their approach is complementary or different from yours.

Conclusions: Finally, present your conclusions¹. This is similar to the introduction, except that now you have the luxury that you can assume that the reader has read the whole report, and you may thus present your conclusions in a more technical way. This is where future work is also discussed. Keep in mind that future work should describe work which could be an interesting scientific/engineering project in itself, and not just a list of minor improvements a code-monkey can do. Examiners are sick of seeing ‘*Improve the user interface*’ as future work.

Acknowledgements: If you want to thank anyone, it can go here.

3 Style Guidelines

Here are some style guidelines which you should follow when writing your report:

1. The maximum length report is 35 pages or 15,000 words (whichever is reached first) inclusive of figures, tables and bibliography but not the title page, the long abstract, table of contents and the lists of figures and tables (if included). Any material beyond the first 35 pages or 15,000 words will not be taken into account by the examiners,

¹And note the plural, unless you have reached just one conclusion.

so make sure you stick to this limit. Trying to win space by changing margins, font size, line spacing, etc. is not allowed and may result in failing the project.

2. You may cite papers by using the L^AT_EX `\cite{aaa}` command, to obtain [?]. Citing multiple papers may be done using a single `\cite{aaa,bbb}` command to produce a reference like [?, ?].
3. Always capitalise references to concrete sections, tables and figures — see, for example, Section 1, Table 1 and Figure ??.
4. Capitalise all words in sections and subsections except for articles, prepositions and conjunctions. If you do not know the difference between a preposition and a proposition, I suggest you look the terms up in a grammar book.

4 Conclusions

There is not much to conclude here.

Acknowledgements.

I would like to thank Muxxu, my hamster, for his continuous support and for keeping me company throughout this past year. Thanks must also go to my cat for refraining from eating Muxxu. Finally, my supervisors were also occasionally helpful.