

UCC Computer Science   
Final Year Project Report 2008   
  
What is This? A Project for Ants   
  
Student Name: Jake Walsh   
Student Id: 113380396   
Supervisor: Derek Bridge  
Second Reader: TBD  
  
B. Sc. Final Year Project Report   
  
  
  
2016

**Declaration - All my own work**  
  
I declare that, unless otherwise stated,   
this report and the project described is entirely my own work.   
  
Signed: 

**Optional Pages - Dedication & Acknowledgments**Acknowledgments- e.g. Thanks to ..(friends, systems guys etc..) for their suggestions, help, guidance and encouragement,(and the splendid isolation and utter frustration emphasising the ever present risk of failure being the supreme incentive.. no don't say that, although you might want to!)

**Abstract -** highly desirable to essential.   
A short precise summary (~ 5 - 30 lines, absolute max one page side entitled**Abstract**) of the topic and your approach and implementation, basically indicating clearly and concisely the subject area, its importance, interest or relevance, any noteworthy results obtained and their general or specific relevance.   
  
e.g A comparison of search algorithms for... was undertaken, which is of interest to applications in the general area of... which is of importance to resource discovery in... The results indicate a favourable comparison with ... techniques and results.. .but show that the .. technique is particularly well suited to ...   
  
Basically you are addressing the questions:-   
what is this about?; why it is important, interesting or relevant? and what was done?;   
within the context of, but not fully addressing directly, the subsidiary question:   
why should anyone read it let alone read do it?, which is expounded fully in the introductory chapter.   
  
  
**Table of contents (TOC)** - required.to help the reader see the structure and flow of your report at a glance,   
and of course to find his way around, should (s)he need to read a section again.   
  
**Chapters**..   
- sections - appropriate to the main issues of the chapter e.g. general approaches / specific technologies   
- subsections - appropriate to specific approaches or implementations within either above   
  
(and as a 3-level tree, that should be sufficient...)   
  
  
Chapter headings follow, but please note that these are indicative of the thematic content of each chapter, not the title, which should be relevant and appropriate to the topic! The guidelines are general and are not to be followed mindlessly. In cases where some of the delineations between chapters would be artificial, you should take liberties with chapter borders, but follow the general logical sequence. Guidelines on structuring your report are in the previous page - but the following paragraph is copied from that page.   
  
 **Content: Context, Critique and Contribution.**   
Regardless of structure, it is really the content and analysis that counts, and in all sections, from introduction to conclusion, it is in your interest: to place things in context, showing that you know the area in general; to give a critique to show you know the shortcomings or how it could be improved; and finally your contribution (although it does not have to be strikingly original or significant, but must be your own) to show your and have made reasonable design and implementation decisions based on your intended direction within the constraints of resources, be they computing, software or time. Finally, in the conclusions section, you may summarise your own work within similar headings, and leaving the final analysis and critique of you work to then, followed by the most likely contribution any future work could make.   
  
So the pattern for each chapter follows the headings above, but for further insight you should consult the section on grading.   
  
Do not hesitate to mention any major obstacles or shortcomings, particularly if you found a neat way around them - which adds spice to a good write-up!   
  
 **1 - Introduction.**The aim is to place the project in context of it's importance and relevance to those like-minded in science and society, briefly showing any shortcomings or opportunity for improvement, which forms the basis and motivation for your project. You should conclude the chapter with an outline of the structure of your argument as presented chapter by chapter in the remainder of the report.   
  
  
**2 - Background**: Review of existing theory/technology etc. basically describing (and showing you know and have researched) the current state of the art. This is quite important in that it sets the context for your work, but more importantly for you, it shows you know that context. Throughout, and particularly towards the end of each section, or subsection, and especially at the end of the chapter, you may mention any shortcomings of the current approaches which leads naturally to your own rationale for undertaking this project.   
  
In the event that your topic is entirely novel, whether scientifically (unlikely) or technologically (more likely), with no relevant prior background, e.g. to address a new need, occasioned by a new technology, then you may still describe the existing technology, outline the shortcoming which is your opportunity to develop what you did.  
  
If you feel your idea is entirely new, then you may try to outline any foregoing ideas and their shortcomings, which led to the inspiration for your entirely novel approach.   
  
  
**3 - Your approach** - basically it should follow naturally, logically and almost inescapably from your foregoing discussion in Chapter 2. You should also explain and justify your decisions and approach, why it would appear to be the best approach, in the context of other possible approaches. The extent and depth of the approach should be realistic with reasonable fall-back (and spring-forward) positions, allowing for variation in progress. For a software development project, a clear requirements and specification analysis should be outlined, but not necessarily given in full detail.   
  
**4 - Implementation**The aim is to discuss more of the science and art of implementation, such as the design and development decisions made and technologies chosen, rather than the details of project management, unless that is a specific area of study. In my opinion, reporting of project management charts should be condensed, dumped to an appendix, or even omitted, but if you think you need to show you had realistic design aims and goals within the time and resource framework, then you may use them as required.   
  
In discussing the implementation in general, it is necessary to present it within the context of other options, indicating any shortcomings and justifying why you choose to do it the way you did, whether in the choice of algorithms or programming technologies.  
You may also point out how you designed tests into your system, whether experimental to observe the behaviour of an approach, or any techniques used to aid software testing during the development phase, such as 'grey box' testing, where the code designed and used to facilitate the design of test cases. However, you can also include logical testing such as 'assertions' in C into the code if desired.   
  
  
**5 - Results & Testing**   
  
You have courses on various methods for testing, but a quick approach is to trace the execution to check the program flows logically as expected (white-box) and monitor the results and trends for a known set of test cases (black-box). It is often informative to use extreme values or ranges of inputs to really test a system.   
  
Assuming the program is valid and behaving as designed, you then need to check if the design is behaving as desired. Try to explain any observed shortcomings or foibles logically and accurately, illustrating any modifications used to correct errant behaviour, and in turn evaluating those. The end result should be a deeper understanding of the either or both the system under study and/or the software system developed, warts and all!   
  
And don’t be afraid to say why it didn’t work or what the obvious weaknesses are - if you don’t the reader might assume you didn’t spot the obvious or haven’t got a full understanding of the area. Of course, he might just be looking for answers too! Whatever the reason, it is good to do a thorough analysis of the negative as well as the positive - it shows a mature, comprehensive and realistic approach when you appreciate the weaknesses as well as the strenghts of the work - it shows you have both depth and width to your knowledge - the reader may even consider you an expert since it seems that you can even speculate!  
  
**6 - Conclusions and Future Work**The aim here is to summarise your work, and you can review it briefly in it's entirety, within the broader context, at seminal, design, implementation and results stages. Now, with the benefit of hindsight, you can perform a plausible critique and with the option of more time, suggest contributions from future work.  
**Appendices A1, A2 etc..**These contain essential supporting details that would disrupt the logical flow of the report, are not central to the main theme, and do not add to the readers enjoyment but are for the zealous, dedicated or misled. Suitable appendix material includes overly-detailed theoretical derivations, project management charts, full listing of code (\*see below) or results in raw form before being presented as graphs or tables etc.   
  
Of course, you may refer to them in the main text, quote their main results, use snippets but reserve the full details for the appendices - basically the boring detail rather than the inspiring high-points of your work!   
 **Bibliography**A good bibliography is a scholarly addition to any report, and indicates you did your groundwork so that your project is built on a sound foundation, and that your arguments are based on evidence and not opinion. It also avoids allegations of plagiarism.   
  
The style of referencing may vary, and vary from numbers to footnotes, and whatever you use, try to be consistent rather than confusing, but I highly recommend the following method.   
  
For published papers, form a mnemonic from either the leading or dominant letters from a persons surname followed by the year of publication; if more than one publication by the same author in the same year is referenced, then append a, b, c etc. e.g. Anon05b, etc. If more than one author was on the paper you may use 'et al', Latin for 'and others', or just insert & before the year.. .Anon&05. If the '&' character is a control character in your text processing facility, use another character such as '+" if you can't override the setting.   
  
The main reason I recommend this is that it is unique and easy to remember and may actually mean something to a cognate reader, let alone the writer, who may remember the author and paper. Moreover, for the writer, there is no need for reordering reference identifiers in the event of insertion or removal of text as occurs for a set of sequential integers (some text processors automatically update the integer set and their mappings for text insertion or removal but not all).   
  
The normal reference entry follows the format, and the list is sorted by reference identifier as used in the main body of the text:   
  
reference mnemonic id   
  
Anon&00 Anonymous, A.; Nameless, N.; Much ado about nothing, Intnl Jnl of the Vacuous Virtuous 10(4), 2000,100-200   
  
  
For a book:   
  
in its entirety: you may replace the Journal Title, page-range etc by Publisher, Place and ISBN No.   
reference mnemonic id Publisher, Place, Year, ISBN   
  
or a section/chapter : you also need to include the book title, page range, book editors (if distinct from article authors),   
  
reference mnemonic id in Publisher, Place, Year, ISBN   
  
  
  
For a website or page, you should give the full URL, and site title, and if you are unsure of the author, choose an informative mnemonic based on the topic or title of the site. eg. GNU etc. Although standards may evolve, custom and practice sets the pace, if not the trend.   
  
reference mnemonic id in   
  
  
**Electronic Resources:Program Listings\*.**You may include code extracts in the main body of the text, to illustrate particularly elegant or neat solutions, or highlight some aspect of your design or implementation, but there is generally little point in including the entire code within the main body of the report.  
Some list the program in it's entirety in a hardcopy appendix, or even in a separate bound booklet, but including a CD inside the back cover seems a much more convenient solution for everyone. Hosting code on a website is not recommended, because websites are all rather transient and somewhat unreliable, and may only add to problems or allegations of plagiarism, even against yourself, despite you being the valid author.