

# **A Short Guide To Dissertation Writing**

## **Abstract**

This guide is meant to help you produce a good dissertation. It gives advice on how to gather relevant material, how to organise it into a suitable form and how to turn it into written prose. It also describes the conventions that should govern the structure of the dissertation and suggests some descriptive devices that you can use to make it more effective. Finally, it guides you through the various phases of your final year project.

## Contents

1.	<b>Introduction</b>	3
2.	<b>Gathering Material</b>	3
3.	<b>Arranging Material and Structuring the Dissertation</b>	3
	3.1 The Introduction	5
	3.2 The "Background" Section	5
	3.3 The "Specification and Design" Section	6
	3.4 The "Implementation" Section	6
	3.5 The "Evaluation" Section	7
	3.6 The "Future Work" Section	7
	3.7 The Conclusions	7
4.	<b>Writing the Report</b>	7
	4.1 Potential Readership	8
	4.2 Identifying commonality	8
	4.3 Sections and Subsections	8
	4.4 Stylistic Conventions	8
5	<b>Useful Descriptive Devices</b>	9
	5.1 Pointers	9
	5.2 References	9
	5.3 Footnotes	10
	5.4 Lists	10
	5.5 Figures	10
	5.6 Literal Text	11
6	<b>Supporting Structures</b>	11
	6.1 The Title page	11
	6.2 The Abstract	12
	6.3 The Acknowledgements	12
	6.4 The Table of Contents	12
	6.5 The Glossary and the List of Abbreviations	12
	6.6 The Appendices	12
	6.7 The Bibliography	12
7	<b>Sources of Further Guidance</b>	13
8	<b>Conclusions</b>	14
9	<b>Stages in the Submission of a Final Year Project</b>	15
	<b>Bibliography</b>	16
	<b>Appendices</b>	17

## 1. **Introduction**

This guide is meant to help you produce a good dissertation. A good dissertation is one that presents your project work effectively. It should contain all kinds of material relevant to the project, including background information; it should be organised into a logical framework; and it should be supported by textual structures, which should follow well-established convention consistently.

The guidelines here are arranged roughly in the order that you will need to use them. Section 2 gives advice on collecting suitable material. Section 3 describes the overall structure of a dissertation, distinguishing between its main body and the supporting structures. This section goes on to describe the more detailed structure of a typical computing dissertation, suggesting that you organise your material in the same way. Section 4 gives general advice on how to write the main body. Section 5 suggests various devices such as figures, tabulated lists and so on which you can use to improve the effectiveness of your writing. Section 6 describes the conventions governing the structures which support the main body, while Section 7 suggests sources of further guidance. Finally Sections 8 and 9 gather our final recommendations and a practical guide through the various phases of your final year project.

The appendices also attempt to make this guide as complete as possible. Appendix A includes the style guide or regulations which govern dissertation format in accordance with the CSAI department and the University of Malta. These dictate various details such as what line spacing and font sizes you can use and students should abide to them before you print and bind the complete dissertation. Appendix B shows a sample Title page to give you an indication of what to include on this very first page. Finally, Appendix C gives a template of the Inception report as required in one of the stages in the submission of your final-year project (see Section 9).

## 2 **Gathering Material**

First we will identify the kind of material you need to collect before you can begin to write in earnest.

Your dissertation is primarily a report of your project and so most of the material will consist of your own ideas and experiences gained while carrying out the project. While you are working on the project keep a notebook handy and record all relevant information. Typically such information will be

- specifications,
- designs,
- results of tests,
- notes from meetings or interviews with,
  - your supervisor,
  - potential end-users,
  - technical experts,and so on.

Also, keep a diary of all your project-related activities. This will show progress made during the life of the project and provide a record of how you spent the time. More specifically, when you are testing and debugging programs, keep a running log of you debugging activities and their results. You will then have a record of the unforeseen difficulties you met and solved. Summaries of these may be well worth including in the dissertation (see section 3.4).

It is very important that you supplement the material you generate yourself with relevant material from other sources. A good dissertation will show that you are aware of relevant work that other people have done (see section 3.2 for details). You should make every effort to include references to such work in your dissertation. References to work in periodicals (i.e. magazines and journals) tend to be more useful than references to text books, as periodicals are usually more specialised and up to date. References to technical manuals can also be included.

### 3 Arranging Material and Structuring the Dissertation

Once you have started to gather material you can begin to arrange it in a form, which can then be refined into the final dissertation.

All good technical dissertations, whatever their subject, follow certain well-established conventions and have a similar overall shape. They all consist of a main body surrounded by other structures, which support the main body in various ways. Some of these are mandatory, others are optional.

Figure 3.1 shows the layout we are describing. For the moment we will concentrate on the main body of the dissertation and leave the supporting structures until later. We recommend that you do the same when writing your dissertation.

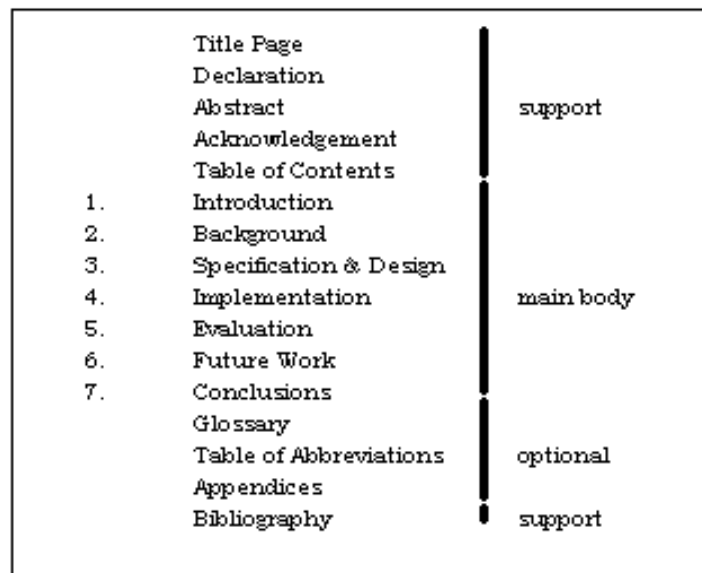


Figure 3.1 Computing Dissertation Structure

Dissertations describing computing projects tend to have a main body with a characteristic structure. This is especially true for projects whose aim has been to develop particular software systems. The body is typically divided into seven major sections under the headings:

- Introduction
- Background
- Specification and Design
- Implementation
- Evaluation
- Future Work
- Conclusions

We will look at each of these in more detail shortly. Assuming your project aims to develop some software system, you can use this characteristic structure as a kind of template for partitioning the material. You can then refine each part to produce a first draft of the main body. Of course, your project may not fit precisely into this pattern. For example, its aim may be to produce not a finished system, but only a specification of the system's requirements along with some prototypical code showing how these might be implemented. Nevertheless, the characteristic structure described here should still strongly suggest the kind of material that your dissertation should contain and how it should be organised.

### 3.1 *The Introduction*

A good introduction should give the reader an overview of the project without assuming special knowledge and without introducing any specific material that might obscure the overview. It should anticipate and combine the principal points that are described in more detail in the rest of the dissertation. Normally, it should include such things as

- ❑ the aim(s) or goal(s) of the project.
- ❑ the approach used in carrying out the project,
- ❑ any assumptions on which the work is based,
- ❑ an overview of the dissertation.

The overview of the dissertation can simply take the form

*In section 2 of this dissertation we describe ... In section 3 we go on to ...*

The limitations of the project, as well as its achievements, can be discussed in the introduction too. These could include the reasons for any disparity between the original aims of the project and what was actually achieved.

### 3.2 *The "Background" Section*

The purpose of the Background section is to provide the typical reader with information that they cannot be expected to know but which they will need in order to fully understand and appreciate the rest of the dissertation (see section 4.1 for details of who a typical reader might be). This section may describe such things as

- ❑ the wider context of the project,
- ❑ the anticipated benefits of the system,
- ❑ the likely users of the system,
- ❑ any theory associated with the project,
- ❑ the software development method(s) used,
- ❑ any special diagramming conventions used,
- ❑ existing software (or hardware) that is relevant to the system, and so on.

The wider context of the project includes such things as its non-computing aspects. So, for example, if you are producing software for a specific organisation then you should describe the aspects of that organisation's business that are relevant to the project.

Relevant existing software systems that you should mention could be ones that, for example:

- are similar to the one you are building,
- support your project in some way,
- your project aims to extend.

You need only describe things that will be unfamiliar to the potential reader. Your project will almost certainly use all kinds of existing software such as language compilers, subroutine libraries, etc. but you can assume that the reader will be fully acquainted with, for example, general purpose programming languages such as Pascal, C, Fortran, Basic, etc., and the better known specialised packages such as ORACLE, INGRES, Lotus 123, etc., You should mention the particular variety and possibly version number, e.g. Turbo Pascal 5.0, but you need say nothing more than that.

If your project depends on any esoteric software such as specialised subroutine packages or any of the more obscure fourth generation languages, you should describe it briefly and discuss whatever features are relevant to your project. Often this can be done by comparing it to some well-established piece of software.

For example

**The Descartes language is like a restricted version of Pascal but with the following extra features...**

Other background information could consist of the sequence of events leading up to the present project or the results of earlier investigations. You could also discuss such things as any cost or time constraints imposed on the project.

### *3.3 The "Specification and Design" Section*

The purpose of the Specification and Design section is to give the reader a clear picture of the system you have created and why you created it in the way that you did.

Describing a software system effectively usually means describing it from more than one viewpoint. Each viewpoint will convey some information about the system that other views omit. (You would use the same technique when describing any complicated construction such as building, an aircraft, a novel or a painting.)

Possible viewpoints might be

- ❑ the system as experienced by a potential end-user,
- ❑ the dynamic behavior of the system,
- ❑ how data flows through the system,
- ❑ what data types are implemented in the system,
- ❑ the static architecture of the system, i.e. how the code is partitioned into modules, etc.

A common approach is to describe first the static architecture, identifying modules and groups of closely connected modules, and then to apply other views to each of these groups. Fine details, specifically details of code, should be left out. Also, any complete rigorous specification (assuming, optimistically, that you have had the time to produce one) is better relegated to an appendix.

We strongly recommend that you make extensive use of diagrams such as EARDs or STDs, or other pictorial techniques (see Section 5.5 for more advice on this).

As well as describing the system, it is very important that you *justify* its design, for example, by discussing the implications of different design choices and then giving reasons for making the choices you did. Typically these implications will relate to the aims of the project and to aspects of it discussed in the Background section.

The design of the system will almost certainly have evolved while you were developing it. Obviously you should describe its final state but often there are good reasons for describing intermediate states too; for example, if you want to discuss details of the design method used. If you do this, take special care to make sure the reader does not get confused between different stages of the design.

### 3.4 *The "Implementation" Section*

The Implementation section is similar to the Specification and Design section in that it describes the system but it does so at a finer level of detail, down to the code level. It can also describe any problems that may have arisen during implementation.

Do *not* attempt to describe all the code in the system, and do not include large pieces of code in this section. Complete source code listings should be put in an appendix (see section 6.6). Instead pick out and describe just the pieces which, for example,

- are especially critical to the operation of the system,
- you feel might be of particular interest to the reader for some reason,
- are exemplary, i.e. they illustrate an algorithm, data structure, etc. that is used widely throughout the system.

You should also mention any unforeseen problems you encountered when implementing the system. Common problems are:

- difficulties becoming familiar with existing software, because of, e.g.
  - its complexity,
  - lack of documentation,
  - lack of suitable supporting software,
  - over ambitious project aims.

A seemingly disproportionate amount of project time can be taken up in dealing with such problems. The implementation section gives you the opportunity to show where that time has gone.

### *3.5 The "Evaluation" Section*

In the Implementation section you should describe to what extent you achieved your goals.

You should describe how you demonstrated that system works as intended (or not, as the case may be). Include comprehensible summaries of the results of all critical tests that you made. You might not have had time to carry out any fully rigorous tests - you may not even have got as far as producing a testable system, but you should try to indicate how confident you are about whatever code you have produced, and also suggest what tests would be required to gain further confidence.

You must also critically evaluate your system in the light of these tests, describing its strengths and weaknesses. Ideas for improving it can be carried over into the Future Work section.

### *3.6 The "Future Work" Section*

It is quite likely that by the end of your project you will not have achieved all that you planned at the start; and in any case, your ideas will have grown during the course of the project beyond what you could hope to do in the time available. The Future Work section is for expressing your unrealised ideas. It is a way of recording that 'I have thought about this', and it is also a way of stating what you would like to have done if only you had not run out of time. A good Future Work section should provide a starting point for someone else to continue the work which you have begun.

Needless to say, a dissertation where the Future Work section takes up most of the main body will be looked upon sympathetically.

### *3.7 The Conclusions*

The Conclusions section should be a summary of the project and a restatement of its main results, i.e. what has been learnt and what it has achieved. An effective set of conclusions should not introduce new material. Instead it should draw out, summarise, combine and reiterate the main points that have been made in the body of the dissertation and present opinions based on them.

The Conclusions section marks the end of the dissertation proper. Some authors say that it is important to finish on a positive note, irrespective of how well or badly the project has gone.

## 4 Writing the Dissertation

Once you have gathered and organised enough material you can begin to turn it into written prose. To write effectively requires sustained concentration over long periods of time. Even with the incremental authoring possibilities that word processing offers, writing is best done in long uninterrupted sessions. Most people find it difficult and exhausting.

There are rules you can follow which may make the task easier and which will certainly improve the quality of your writing, but unfortunately there are rather a lot of these and in a guide of this size we can only offer a few scraps of general advice.

Our advice is to:

- ❑ keep your potential readership in mind,
- ❑ identify commonality,
- ❑ use sections and subsections,
- ❑ follow stylistic conventions.

More extensive guidelines are given in [Gowers 87] and we recommend very strongly that you get yourself a copy. [Barrass 78] and [Cooper 64] may be helpful. There are also a number of periodicals, for example the IEEE Transactions on Professional Communication, which regularly carry articles related to technical writing. The dissertation's structure does not necessarily dictate the order in which you write it. If you want you can start by writing the introduction, then the Background section, and so on, but this is up to you. Some people find that writing the introduction first gives direction to writing the other sections, but others prefer to leave writing the introduction until last.

### 4.1 *Potential Readership*

Always keep your potential readers in mind and repeatedly review what you have written putting yourself in their place. Look at the draft, sentence by sentence, and ask yourself: 'Will this make sense to the readers given their existing knowledge and what I have told them up to now.' In the case of your dissertation you can consider the potential readership as

- ❑ your academic supervisor,
- ❑ the external examiner (usually a computing professor from another university),

and quite possibly

- ❑ future students.

Also, if your dissertation is of sufficient quality, your supervisor may consider submitting part of it to a journal for publication as a paper, in which case it may eventually be read by a substantial number of computer professionals.

#### *4.2 Identifying Commonality*

You can often both clarify text and reduce its bulk if you can identify generality or commonality among the ideas you are expressing. You can then revise the text so that the common factors are described first followed by details of how specific individual ideas differ from them. (Similar techniques are used in programming to create subroutines and other modules).

#### *4.3 Sections and Subsections*

The main body of the dissertation should be divided into sections, along the lines we suggested in section 3 if these are appropriate. Each section can, if necessary, be divided into subsections, and so on recursively. Such nesting can be used to suggest some kind of hierarchical relationship between sections (such as the general special relationship mentioned in section 4.2). This can become obscure though if the nesting gets to more than about three levels deep. Each major section should begin on a new page. All sections and subsections should be numbered and headed. Numbering should be like this: 3.10.7 - for sub-section 7 in subsection 10 in section 3.

It is important that you start each major section with a few sentences that summarise the rest of the material in it: i.e. tell the reader what you are about to tell them. This has the effect of "softening up" the reader so that when they move on to the body of the section they feel confident about the direction in which you are taking them. They are reassured at regular intervals when they encounter ideas which you have told them to expect. Without the introductory sentences, the overall effects is like a mystery tour of ideas with each new idea coming as a surprise. It is sometimes difficult to appreciate this when you are the author because you are already intimately familiar with the whole route that the dissertation takes.

#### *4.4 Stylistic Conventions*

There are all kinds of stylistic conventions relating to technical writing which you should try to follow. For example,

- use the pronoun 'we' rather than 'I' (even if there is only one of you),
- avoid shortened for such as ' don't and there's',
- avoid colloquialisms,
- and so on

Refer to [Gowers 87] for a comprehensive list. To some extent you can use your own judgment about what conventions to follow.

Whatever you do though, you must *consistent*. Writing where the language style or aspects such as font or character size change arbitrarily can be very distracting for the reader and looks amateurish.

## 5 Useful Descriptive Devices

In this section we will mention some well-established descriptive devices, which you can use in your dissertation to improve its quality.

### 5.1 *Pointers*

Pointers are just references to other parts of the same document. For example,

**This module contains procedures for operating on variables of type WINDOW (see sect 2.2).**

Section numbers will change if sections are added or deleted, so it is a good idea to wait until the report is almost complete before putting in any pointers.

Backward pointers, pointers referring to a section earlier on in the dissertation, can make explicit connections between parts of the document that may not be connected obviously. Forward pointers can be used, for example, to reassure the reader that you are not going to leave them stranded after you have introduced a new idea without explaining it. For example.

**This procedure uses the Volestrangler algorithm (which we describe in section 2.2).**

Note that too many forward pointer are an indication that the report could be organised better.

### 5.2 *References*

In section 2 we said that you should make every effort to relate your work to that of other people. Provided that their work is documented you can enter details of the documentation in the bibliography (see section 6.7) and refer to the entry within you text using some kind of key. It is important that you properly give credit to all work that is not strictly your own and that you do not violate copyright restrictions.

There are various conventions for creating bibliographic keys. For example, you can quote the name of the author and the year of the publication, e.g.

**A more detailed description can be found in (Borland 89).**

There are several variations. For example, some authors prefer to use only the first three or four letters of the name, e.g. (Borland 89). Whatever convention you choose, be consistent.

It can be helpful to the reader if, for books and other publications, you specify the chapter or page number too, e.g. (Borland 89 ch 3) or (Borland 89 p22).

A simpler key convention is just to number each publication in the bibliography and then use this number as the key, for example

**A more detailed description can be found in (3).**

This method has the advantage that it does not cause such an obvious break in the text, especially when several references have to be made at the same place. It has the major disadvantage though that if you want to add a new publication to the bibliography, while keeping the bibliography in order, then you will probably have to remember many of the entries and consequently update all the references to them in the text. This can be a very tedious process, unless you have a clever word processor.

### 5.3 *Footnotes*

Many word processors have facilities for handling footnotes. By all means use them, in particular when you want to make a comment which is not strictly relevant or which would upset the flow of ideas in the text.

### 5.4 *Lists*

Traditionally, collections of items are listed within the text using the adverbs 'firstly', 'secondly', etc. Often though it is clearer to tabulate the items, particularly if there are many of them. The simplest way of doing this is to use a "bullet" list. Various examples of bullet lists appear throughout this guide. Sometimes there is a need to nest one list inside another. To distinguish the two lists, the inner one can be indented and have a dash instead of a bullet. Lists with more than one degree of nesting tend to appear confusing and therefore we do not generally recommend them,

Listed items can be keyed using numbers, letters or other labels, Bibliography entries are an example of keyed items (see section 6.9). It is best to use a key only when it is necessary though.

### 5.5 *Figures*

A dissertation that uses figures (i.e. diagrams or other pictorial techniques) to illustrate ideas will probably be easier to digest than one that does not. We recommend that you use figures wherever appropriate.

Be careful though, when drawing diagrams try to keep to a graphical notation that has been introduced in your course, or that you have seen published widely, and use it consistently. Computer science, unlike most other professions, has few established conventions governing the use of diagrams and this means that diagrams can sometimes make ideas more obscure rather than clarifying them. Look through any recent computing publication, particularly one that contains the work of several authors, and you will see a confusing variety of different symbols and diagrammatic forms. They may illustrate very similar kinds of underlying relationship but often this is not obvious.

If you feel you have to invent your own notation, remember that the best ones are usually the most *economical* i.e. they use only a few different kind of symbols. Also, you *must* declare the precise meaning of your symbols in a key. A very common mistake is to use arrows to illustrate some kind of relationship between items without declaring what the relationship is.

Graphics editors (i.e. picture processors) can be extremely useful, particularly if you have a great deal of drawing to do or if there is a lot of commonality among the drawings (because cut and past operations can then be used with great effect). A graphics editor is incorporated in the word processing part of the Microsoft Works package on the department's Macintosh computers.

All figures should be labeled and captioned like this: Figure 3.10 Sub-System Architecture, for the tenth figure in section 3. The label can be used when you need to refer to the diagram in the text.

## 5.6 *Literal Text*

It is important when writing about software systems to distinguish in the text between the ordinary natural language that you are using and program code or other literal text. If you are using a word processor which offers both proportionally spaced and fixed width character fonts then there is a straightforward way of doing this. Program code and other literal text can be written in a fixed width font such as Courier while the natural language text can be written with a proportionally spaced font such as Times or Geneva. For example:

**The procedure draw\_circle (p:POINT, r:REAL) draws a circle of radius r at point p on the screen.**

Other similar kinds of text, UNIX commands for example, can be treated in the same way. Using a fixed width font for code means that the code appears in the document much as it would do on most computer screens. If you only have fixed width characters available on your word processor then put program code etc., into italics or bold text. Note that using more than a few different character fonts, styles or sizes can make text look very untidy.

## 6 Supporting Structures

In section 3 we said that a dissertation consisted of a main body plus other textual structures that surround and support the body. There are well-established conventions governing the purpose and format of these supporting structures, which we will describe now. The structures are, in order of appearance in the dissertation:

- ❑ the title page, (see Appendix B for an example)
- ❑ the abstract,
- ❑ the acknowledgements,
- ❑ the table of contents,
- ❑ the introduction.

Then comes the main body of the dissertation, and this is followed by

- ❑ the conclusions,

and possibly

- ❑ a glossary,
- ❑ a list of abbreviations,
- ❑ one or more appendices.

and finally

- ❑ the bibliography.

Each of the elements listed above should begin on a new page. All pages should be numbered with page 1 being the first page of the introduction. The pages preceding page 1 should be given Roman numerals (i, ii, iii, iv, etc.). The introduction and the conclusions are considered to be major sections of the dissertation should it be numbered as such (see section 4.3).

### 6.1 *The Title Page*

The title page, as demonstrated in Appendix B, should include:

- ❑ the title of the dissertation,
- ❑ the name of the author,
- ❑ the name of the supervisor and possibly the observer,
- ❑ the name of the department and college,
- ❑ the date of the completion of the dissertation,
- ❑ the qualification for which the dissertation is a part,

The title itself should be short and yet aim to describe the content of the dissertation as accurately as possible.

## 6.2 *The Abstract*

This is a summary of the dissertation. It must be less than 300 words long. It should give enough information to allow a potential reader to decide whether or not the whole dissertation will be of interest to them. It should briefly describe the main features of the dissertation including the aims and conclusions. It should be both self-contained and self-explanatory, and it should refer to anything not mentioned in the rest of the dissertation.

## 6.3 *The Acknowledgements*

This section should be used to record any debt for the use of facilities or help from particular sources. You should mention any organisations that have helped fund you while you have been carrying out the project. Also, if you were a placement student, it would be diplomatic to include the name of your supervisor in the host organisation where you were placed and any of his or her colleagues who helped you.

## 6.4 *The Table of Contents*

The table of contents gives the reader a view of the detailed structure of the report by giving section headings and associated page numbers.

If your dissertation contains many figures or if it refers to the same figure many times, you should consider listing them along with their page numbers in a separate table of figures.

## 6.5 *The Glossary and Table of Abbreviations*

If you use any abbreviations, obscure terms, or esoteric acronyms in the dissertation then their meaning should be explained where they first occur. If you go on to use any of them extensively then it is helpful to list them all in a table at the end so that readers can quickly remind themselves of their meaning.

## 6.6 *The Appendices*

Appendices are repositories for material which you want to include in the dissertation but which would seriously obstruct the flow of ideas put anywhere in the main body. In computing dissertations it is normal to include a printout of (the final version of) the

source code of the system in an appendix. (if you have carried out the project in placement then make sure that you have your host organisation's permission to do this). Other documents you may have written, such as user manuals or formal specifications should go here too.

Appendices should be headed by letters in alphabetical order, i.e. **Appendix A**, **Appendix B**, etc.

### 6.7 *The Bibliography*

The bibliography is a table of publications or other sources of information, which are referred to in the text of the dissertation. Its purpose is to provide the reader with all the information needed to find the appropriate document.

The entries in the bibliography should be arranged in alphabetical order by the name of the author. Each entry should start with the following information

- the name(s) of the author(s),
- the title of the publication.

If there are several authors, their names should be arranged in the order that they appear on the document itself.

If the document is a book, the entry should then state the publisher. If the publisher is not a major one, or if distribution is localised to a particular community then include whatever details are necessary for the document to be located. Including the ISBN number can be useful.

Alternatively, if the document is an article or paper in a periodical, the entry should contain

- the name of the periodical,
- the volume and issue number, or the date of the issue,
- the numbers of the first and last pages of the article or paper.

Every entry, whatever kind of document it refers to, should state the year of publication. It should also be labeled with a key. The format of the key will depend on the key convention you have chosen (see section 5.2). There may of course be publications for which some information is not available. Technical manuals for example, rarely include the names of their authors. In such cases include as many of the above items as possible.

Below are examples of bibliography entries that illustrate some elaborations of the rules we have described.

[Aho et al. 86] Aho A.V., Sethi R., Ullman J.D., *Compilers - Principles, Techniques and Tools*, Addison Wesley, (1986)

[Anson 82] Anson E.K., *The Device Model of Interaction*, *Computer Graphics*, 16(3), 107-114, (1982).

[Borland 89] Borland, *Turbo Pascal Reference Guide, Version 5.0.*, Borland International Inc., (1989)

[Kernighan 84a] Kernighan B.W., Pike R., *Program Design in the UNIX Environment*, *AT&T Tech. J.*, 63(8 pt 22), 195-1605, (1984)

[Kernighan 84b] Kernighan B.W., Pike R., *The UNIX Programming Environment*, Prentice - Hall, (1984)

Common abbreviations are sometimes used, For example, many academic publications have titles such as "Transactions of ..." or "Proceedings of..." and these are often abbreviated to "Trans..." or "Proc..."

Another convention for handling citations and references, (apart from the symbolic conventions illustrated above, is the Numeric convention, shown below. Whichever you choose, it is important to be consistent.

**Numeric.** In this case citations in the text would appear as follows: is fully discussed in Abela et al. [1]. Maltese perspectives (see Abela [1], Zahra [2] also). The bibliography would then look something like this:

[1] Z. Abela, A. Zahra & C. Camilleri, *The Halting Problem: Traffic Accidents in Malta*, *Journal of Unsolved Problems* 12(9):123-136, 1998.

[2] Zahra, *The Maltese Neolithic Temples and Fractal Art*, PhD thesis, Department of Archeology, University of Malta, 1987.

## 7 **Sources of Further Guidance**

We have said several times that this guide is not complete and we have suggested other publications, which will give you further guidance. We also recommend the following.

Textbooks usually demonstrate good technical writing especially if they are produced by a reputable publisher. They will also provide illustration of the use of descriptive devices, etc.

While writing this guide, we have tried to follow the guidelines it contains. Obviously though, a guide like this has a different purpose to that of a dissertation so total adherence has been impossible. We have tried to set an example but it is not perfect.

Finally, there will be specific aspects of your dissertation that only your academic supervisor can advise on. It is very important that you discuss an outline of the dissertation with him or her before you begin to write up.

## 8 Conclusions

These are our main recommendations.

- ❑ Record all relevant information generated by the project;
  - use a notebook,
  - keep a diary,
  - log debugging sessions.
- ❑ Gather further material from publications or other external sources
- ❑ Organise the material under the headings
  - "Background",
  - "Specification and Design",
  - and so on.
- ❑ Turn this material into written prose to form the dissertation's main body.
- ❑ When writing the main body
  - keep your readership in mind,
  - identify commonality,
  - use sections and subsections,
  - follow stylistic conventions consistently.
- ❑ When appropriate use
  - pointers,
  - references'
  - figures,
  - and other descriptive devices.
- ❑ Produce all required supporting structures according to convention, after completing the main body.
- ❑ For examples to follow, look at
  - text books from reputable publishers,
  - the way this guide is written
- ❑ Discuss an outline of the dissertation with your academic supervisor before you begin to write up.
- ❑ For examples to follow, look at
  - text books from reputable publishers,
  - the way this guide is written
- ❑ If your report grows considerably longer, it is advisable to discuss the issue with your supervisor before submitting it.
- ❑ Note that submitted copies of the report are to be hardbound with your name, degree for which the project has been submitted and year of submission written on the spine.

## 9 Stages in the Submission of a Final Year Project

*8.1 Proposals:* During the summer recess the department will publish a list of project proposals which students are to choose from. If you have any ideas of your own it is up to you to write a proposal and find a member of staff willing to supervise your project.

*8.2 Preparation:* The summer recess is a good opportunity to carry out some basic groundwork on the project, bearing in mind the inception report. Most students familiarise themselves with the background literature during this phase. Your supervisor will provide you with some pointers, but be prepared to take on research of your own.

*8.3 Inception Report:* Just after the Christmas recess you will be asked to present an inception report. By then you should have a clear idea of where your project is headed. Include a detailed project proposal, what parts have already been completed, a tentative timetable to which work should adhere, a tentative chaptering of your report and a bibliography. This report is to be approximately 3-5 pages long. This report should be read and signed by both supervisor and observer and registered. The Departmental Secretary should keep a log of inception reports that have been signed. See Appendix C for a sample inception report.

*8.4 Project Synopsis:* At the beginning of May you are to hand in a **ten page synopsis** describing the project aims and achievements. This synopsis should be about 10 pages long and will be forwarded to the external examiner. Note that this synopsis is the first impression the external examiner will get of your project. It is thus very important that you do not take it lightly.

*8.5 Report First Draft:* You should have a first draft finished by the end of April. Your supervisor will give comments and recommendations as to how it can be improved over the next two weeks.

*8.6 Presentation:* a part of the assessment consists of your ability to present the project before a live audience. Criteria of assessment include clarity (your ability to present the project clearly), knowledgability (familiarity with the subject and its background; ability to answer questions), and quality of materials used for presentation (the medium should suit the message). The oral presentation takes place around the middle of May.

*8.7 Report Final Draft:* Two weeks before exam week (the exact date varies from year to year) you will be asked to submit three bound copies of the report to the department.

Remember that the **submission date is fixed** and *no extensions can be granted*.

## **Bibliography**

[Barrass 78] Barrass R., *Scientists Must Write*, Chapman & Hall, (1978)

[Cooper 64] Cooper R.M., *Writing Technical Reports*, Penguin (1964)

[Fortune 86] Gowers E.G., (revised by Greenbaum & Whitcut), *The Complete Plain Words*, 3rd ed., Penguin, (1987)

## Appendix A: Style Guidelines

<b>Paper Size</b>	A4
<b>Printing</b>	One side of the paper only
<b>Line Spacing</b>	1.5
<b>Font Size</b>	12 (some sections in 10pt OK)
<b>Top, Bottom, Left Margins</b>	3cm
<b>Right Margin</b>	2 cm
<b>Page Numbering</b>	Arabic numerals, bottom of page
<b>Maximum Length</b>	Approx. 30K words (= 100 pages)

Appendix B: Sample Title Page

**Intelligent Systems for the Automatic Synthesis of  
Undergraduate Dissertations: An AI Experiment**

The Author

Supervisor: The Supervisor

Observer: The Observer



**Department of Computer Science and AI  
University of Malta**

**June 2002**

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

## **Appendix C: Structure of Inception Report**

The inception report should include at least the following elements:

**Title of Project**

**Supervisor**

**Observer**

**Description**

- Background
- Aims
- Methods
- Deliverables

**Timetable**

- Milestones
- Implementation Plan

**We hope this guide is of some use. Good Luck.**