Project Title

Author Name

September 201x

Dissertation submitted in partial fulfillment for the degree of

Master of Science in *(insert your degree title)*

Computing Science and Mathematics

University of Stirling

Abstract

Summary of the dissertation ***within one page***.

This template starts the page numbering at the foot of this page. While you are printing drafts, you might find it useful to add the printing date and time into the footer – to help you, and your supervisor, tell which version is most current.

Note: You are required to submit one extra copy of your title page and Abstract.

It is suggested that the abstract be structured as follows:

- Problem: What you tackled, and why this needed a solution
- Objectives: What you set out to achieve, and how this addressed the problem
- Methodology: How you went about solving the problem
- Achievements: What you managed to achieve, and how far it meets your objectives.

Attestation

I understand the nature of plagiarism, and I am aware of the University?s policy on this.

I certify that this dissertation reports original work by me during my University project except for the following (*adjust according to the circumstances*):

- The technology review in Section 2.5 was largely taken from [17].
- The code discussed in Section 3.1 was created by Acme Corporation (`www.acme-corp.com/JavaExpert`) and was used in accordance with the licence supplied.
- The code discussed in Section 3.5 was written by my supervisor.
- The code discussed in Section 4.2 was developed by me during a vacation placement with the collaborating company. In addition, this used ideas I had already developed in my own time.

Signature: *(you must delete this, then sign and date this page)*   **Date**

Acknowledgements

Acknowledge anyone who has helped you in your work such as your supervisor, technical support staff, fellow students or external organisations. Acknowledge the source of any work that is not your own.

How to Create Table of Contents and List of Figures

The table of contents on the following page is automatically generated from the `\part`, `\section`, `\subsection`, etc., commands in the `.tex` file. Depending on your configuration, you may have to typeset your document twice in order to update this section (the first typesetting will create a `.toc` file, which is then used during the second typesetting).

Similarly you can automatically generate a list of figures from the `\begin{figure}` and `\end{figure}` commands in this document. Again, you may need to typeset your document twice in order for this list to update.

In your document, remove this entire section and only leave the lines

```
\tableofcontents
\listoffigures
```

These automatically begin new pages that look like Chapters in the finished pdf.

Contents

List of Figures

1 Introduction

For editorial consistency, it is important to use LaTeX formatting properly. To use italics, replace the dots in \textit{...} with the text you would like to be italicized. To make text bold, use \textbf.

Chapters are entered using the \chapter command. For example, this chapter begins with \chapter{Introduction} in the .tex file. The chapter command automatically moves to the start of a new page and supplies the next chapter number. The first paragraph after a heading automatically will have no indent on the first line. In the .tex file, it does not matter whether you leave a blank line after the \chapter command or not – the typeset pdf will have it anyway.

The remaining paragraphs will have an indent.

In general, use the default spacing that headings and paragraphs give you. Avoid using new-lines or spaces to format text. If you need to use quotes, preferably use single curly quotes '...'. In the .tex file, the opening quote is typed as a grave accent (`) and the closing quote is typed as an apostrophe ('). If you wish to emphasise something, usually use italic font.

Remember to save frequently while you are working!

1.1 Background and Context

Give the background to your project and context of what you have done. Sections within a chapter are entered using the \section command, which automatically supplies the next section number.

1.2 Scope and Objectives

Define the scope and objectives of your project.

1.3 Achievements

Summarise what you have achieved.

1.4 Overview of Dissertation

Briefly overview the contents of what follows in the dissertation.

2   State-of-the-Art

Summarise current knowledge and what others have done in the various topics of your dissertation – in the application area and in the various technologies that you might have used or did use. Write for someone familiar with computing, but not necessarily expert in the particular topics of your project. Give references to other work by using cross-references to entries in the References section, like this [2].

3 Technical Chapters (change this to something appropriate)

Note: This part of the dissertation will normally be expanded to be a *series of chapters*.

The technical body of the dissertation consists of a number of chapters (just one here, but there will usually be more). Follow a logical structure in how you present your work. This will usually be the phases of the software development cycle, the modules of your system, etc. **However, please do not write your dissertation to read like a diary.**

Include a chapter demonstrating what you have achieved and how your system is used in practice – for example showing a typical session as a series of pasted in screen shots, with an accompanying commentary.

You **should** also include a chapter explaining how you obtained feedback from your "customer" or potential users of your system, what feedback you actually obtained, and your analysis and comments.

3.1 First Section

Subdivide your text into sections, using the \section command.

3.1.1 First Subsection

If necessary, also use subsections. Subsections are entered using the \subsection command (all these heading styles are self-numbering).

3.1.2 Second Subsection

And, as required, more subsections.

3.2 Bulleted and Numbered Lists

Note: This section begins with the code `\section{Bulleted and Numbered Lists}` in the `.tex` file.

Bulleted or numbered lists are entered using the `itemize` and `enumerate` environments, respectively. An **environment** in LaTeX is a block of code in between a `\begin` and `\end` command. For example, the code

```
\begin{itemize}
    \item Up
    \item Down
    \item Left
    \item Right
\end{itemize}
```

would produce the following list:

- Up
- Down
- Left
- Right

The indentation is not necessary (the pdf will look the same even it the `.tex` file does not use indents), but it helps make the code easier to read.

If the `enumerate` environment is used instead, the bullets are replaced by numbers. For example, the code

```
\begin{enumerate}
    \item Up
    \item Down
    \item Left
    \item Right
\end{enumerate}
```

produces the list

1. Up
2. Down
3. Left
4. Right

3.3   Figures and Captions

As an example of a figure, consider Figure 3.1. Captions are entered using the `figure` environment (read the previous section for information about environments in general). The code

```
\begin{figure}[h]
    \center\includegraphics[width=12cm]{image.jpg}
    \caption{Highly Technical Diagram}
    \label{mylovelydiagram}
\end{figure}
```

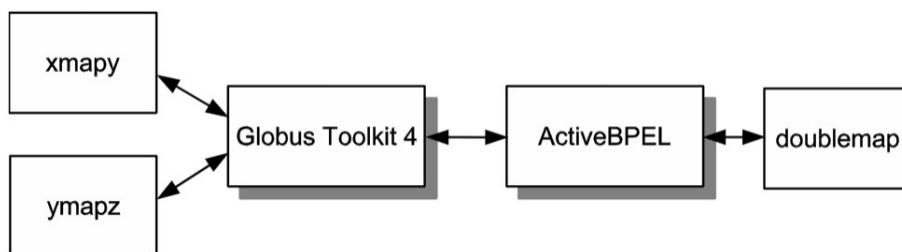will produce the following figure **if the file *image.jpg* is in the same folder as your** `.tex` **file**.

Figure 3.1: Highly Technical Diagram

The [h] direction after the beginning of the environment causes the figure to be placed "here" in the text (at least approximately – sometimes TEX will move the figure slightly if the spacing does not work well in exactly the given location). For large figures, use [t] or [b] instead to place the figure at the top or bottom of a page. You can also leave off the [h] entirely to have TEX make its best guess for where the figure should go.

The \includegraphics command puts an image file from your computer into your finished pdf. **If there is no file with the given name in the folder with your .tex file, your document will not compile at all.** The bracket text [width=12cm] is optional; without it, TEX will use the normal size of the image. Sometimes this will be far too large, so it is a good idea to specify a width directly.

Figures have automatic numbering, and it is possible to make cross-references to figures. The code \ref{mylovelydiagram} will create a link "3.1" in the text with the number of that figure. You can change the text "mylovelydiagram" to be anything you want – it never appears in the final pdf.

3.4   Source Code

To include programming source code in your document, use the `lstlisting` environment.
The LaTeX code

```
\begin{lstlisting}[language=Python, frame=single]
    def factorial(n):
        if n == 0:  return 1
        else:  return n * factorial(n-1)
\end{lstlisting}
```

produces the following in the pdf:

```
def factorial(n):

        if n == 0: return 1

        else: return n * factorial(n-1)
```

You can change `language=Python` to `language=Java`, etc., for different programming languages. The `frame=single` can be removed if you do not want the border around your code snippet. See `https://en.wikibooks.org/wiki/LaTeX/Source_Code_Listings` for syntax coloring and other options.

4 Conclusion

4.1 Summary

Summarise what you have achieved.

4.2 Evaluation

Stand back and evaluate what you have achieved and how well you have met the objectives. Evaluate your achievements against your objectives in section 1.2. Demonstrate that you have tackled the project in a professional manner.

The previous paragraph demonstrates the use of automatic cross-references: The "1.2" is a *cross-reference* to the text in a numbered item of the document; you do not type it as 1.2 but by using the \ref command. The number that appears here will change automatically if the number on the referred-to section is altered, for example, if a chapter or section is added or deleted before it. Cross-references to section are entered with the \ref command just like for figures. The TeX code above reads

```
Evaluate your achievements against your objectives
in section \ref{objectives sec}.
```

For this to work, the code for the text on page 2 must read

```
\section{Scope and Objectives} \label{objectives sec}
```

As with figure labels, the text inside of \label and \ref never appears in the final pdf;

you can make it whatever you want as long as you use the same text in each to complete the reference.

4.3   Future Work

Explain any limitations in your results and how things might be improved. Discuss how your work might be developed further. Reflect on your results in isolation and in relation to what others have achieved in the same field. This self-analysis is particularly important. You should give a critical evaluation of what went well, and what might be improved.

How to Create a References section

Although it looks like a Chapter in the finished pdf, the References section is created using an environment called `thebibliography` rather than a `\chapter` command. Each item in your bibliography should be typed in the `.tex` file as its own paragraph that begins with the `\bibitem` command. The first reference below is typed as

```
\bibitem{GW} Greene, D. and Williams, P. C. \textit{Linear Accelerators
for Radiation Therapy}, Second Edition.  IOP Publishing Ltd., Bristol
and Philadelphia, 1997.
```

Books [1], standards [2], reports [3], journal articles [4], conference papers [5], and web pages [6] are conventionally presented in slightly different ways. Put urls in the `\url` command to give them fixed-width formatting and an automatic link.

To cite an item from your bibliography in the text, use the `\cite` command, which essentially pairs with `\bibitem` the same way that `\ref` pairs with `\label`. For example, the previous paragraph begins

```
Books \cite{GW}, standards ...
```

Bibliography

[1] Greene, D. and Williams, P. C. *Linear Accelerators for Radiation Therapy*, Second Edition. IOP Publishing Ltd., Bristol and Philadelphia, 1997.

[2] ISO. *Language Of Temporal Ordering Specification*, ISO 8807, International Organization for Standardization, Geneva, 1989.

[3] Jacobson, J. and Andersen, O., editors. *Software Controlled Medical Devices.* SP Report 1997:11, Swedish National Testing and Research Institute, Sweden, 1997.

[4] Turner, K. J. The Rules for Sailing Races on PDAs, *J. Navigation*, 23(5):114-240, May 2002.

[5] Ji, H. and Turner, K. J. Specification and Verification of Synchronous Hardware using LOTOS. In Wu, J. Chanson, S. T. Gao, Q. editors, *Proc. Formal Methods for Protocol Engineering and Distributed Systems* (FORTE XII/PSTV XIX), pages 295-312, Kluwer Academic Publishers, London, UK, October 1999.

[6] University of Stirling. Computing Science and Mathematics Research Home Page, `http://www.cs.stir.ac.uk/research`, April 2002.

Appendix 1

You may have one or more appendices containing detail, bulky or reference material that is relevant though supplementary to the main text: perhaps additional specifications, tables or diagrams that would distract the reader if placed in the main part of the dissertation. Make sure that you place appropriate cross-references in the main text to direct the reader to the relevant appendices.

Note that you should **not** *include your program listings as an appendix or appendices.* You should submit one copy of such bulky text as a separate item, perhaps on a disk.

Appendix 2 – User guide

If you produced software that is intended for others to use, or that others may wish to extend/improve, then a user guide and an installation guide appendices are *essential*.

Appendix 3 – Installation guide

If you produced software that is intended for others to use, or that others may wish to extend/improve, then a user guide and an installation guide appendices are *essential*.