

A L^AT_EX TEMPLATE FOR THESES AND DISSERTATIONS

A Dissertation
Submitted to the
Temple University Graduate Board

in Partial Fulfillment
of the Requirements for the Degree of
Doctor of Philosophy

by
James Brodovsky
April 2025

Examining Committee Members:

Jane Doe, Department of Mechanical Engineering (Advisor)
John Smith, Department of Computer Science
Emily Black, Department of Mechanical Engineering
Robert Johnson, Department of Electrical Engineering

©
Copyright
2025

James Brodovsky
All rights reserved

Abstract

This is a template for writing a thesis or dissertation using L^AT_EX. It is designed to help you create a document that meets the formatting requirements of your university. The template includes examples of how to format various sections of your document, including the title page, abstract, acknowledgments, table of contents, list of figures, list of tables, and main body. It also includes examples of how to format figures, tables, equations, and references. This template is intended to be a starting point for your thesis or dissertation and can be customized to meet your specific needs.

Acknowledgments

Leslie Lamport, for making \LaTeX possible. James Yu for creating the fantastic LaTeX Workshop Visual Studio Code extension. The Mathematics department of Temple University, for at, the very least, inspiring me to do this.

To any and everyone who needs this.

Contents

Abstract	iii
Acknowledgments	iv
Dedication	v
1 Introduction	1
2 Background	3
2.1 Why?	3
2.2 Goals	3
2.3 Some General Usage Notes	4
3 Methods	8
3.1 Bibliography and Citations	9
3.2 Block quotes	9
3.3 Figures and Tables	10
3.4 Definitions, Lemmas, and Theorems	11
3.5 Equations	12
3.6 Conclusions	12
4 Headings examples	13
4.1 Section: This heading should be Flush Left, Boldface, and Title Case.	13
A Appendix Formatting	15
A.1 Appendix Section	15
A.1.1 Appendix Subsection	15

List of Figures

3.1	An example figure.	11
-----	----------------------------	----

List of Tables

3.1	Heading styles and their descriptions. Table captions precede the table.	9
3.2	An example table.	11

Chapter 1 Introduction

L^AT_EX is a typesetting system that is widely used for producing scientific and technical documents. It is particularly well-suited for documents that contain complex mathematical equations, tables, and figures, but it is also useful for documents that need to separate out content from formatting. This template (provided in the accompanying `.cls` file) is designed to help you create a thesis or dissertation that meets the formatting requirements of your university, and is specifically tailored to the style guide of Temple University. This file (`thesis-dissertation.cls`) defines the `thesis-dissertation` class, which is a drop-in replacement for the standard L^AT_EX classes and builds off the built-in `report` class, which is used primarily for its numbering rules for things like equations and for the abstract environment.

Because of how L^AT_EX works, this template is effectively in two parts: the class definition and this example implementation. This file is effectively a tutorial and a sort of boot-strapped example. In the same way that compilers for programming languages are often written in the same language they compile, this document is an example of the formatting and structure of a thesis or dissertation and will both document and demonstrate how to use the unique features of the class and how to use L^AT_EX in general. Consider it both an introduction to the class and a light-weight tutorial on how to use L^AT_EX.

In general, you should not need to modify the class file in order to use this. It is designed to be a drop-in replacement for the standard L^AT_EX classes, but with the added benefit of having a template that meets the formatting requirements. Only

delve into the class file if you need to make changes to the formatting that are not covered by the options in this file.

Additionally, this tutorial is set up with *minimal* dependencies. It is designed and intended to be as simple as possible. The additional packages used are noted in the imports section with specific comments about what they are used for in this tutorial. Feel free to remove them when starting your own document. Everything you need formatting-wise is contained in the class file, and for illustration purposes in this file there are references contained in `references.bib`.

For instance, you've probably noticed some coloration or highlighting over certain words (namely in the table of contents). This is done with the `hyperref` package and is primarily used in this document to make URL links accessible by imbedding them in the text as hyper references (like a typical web page). It also has the added benefit of making clickable links to sections, figures, and other such items in the document. While links are not expressly prohibited by the style guide, you should be careful of how it formats the text. Namely, it can color links to something other than black which is the only permitted font color. For the sake of this document, that will be the one exception made to the prescribed style guide.

Chapter 2 Background

2.1 Why?

There are numerous similar templates out there for various other schools. However, they are all very customized and very school specific. One thing that I do specifically like about Temple University's template is that it is very plain and generic and easily reusable in other contexts. The mathematics department even provides a base [L^AT_EX template](#)! However, that version does not appear to be actively maintained, is internally inconsistent, and out of date with the current [handbook](#). There are a few other repos out there that have attempted to tackle this, but they too are actually out of date and unmaintained with the current style guide. On top of that, the other examples I've found for specific TU implementations have been far too author, school, or department specific and would be cumbersome to adapt.

2.2 Goals

The goals of this project are simple. One, provide a truly generic implementation of a LaTeX document template that can be used either for a Master's thesis or Ph.D. dissertation across any department or college within Temple. Two, optionally at any other university as well, provided they follow Temple's style guide or something similar. Failing that, this should serve as an easily adaptable template class that can be forked.

In addition to those general project goals, this tutorial is also intended to be a sort

of introduction to \LaTeX and how to use it. It is not intended to be a comprehensive guide, but rather a starting point for those who are either new to \LaTeX and don't require much in the way of advanced typesetting, just something that meets the style guide and is useful for large documents with cross-references and citations, or advanced users who know their way around who are looking for a simple template to use and need a simple primer.

2.3 Some General Usage Notes

This template and tutorial was built using the TeXLive distribution of \LaTeX and the Visual Studio Code editor with the `LateX Workshop` extension. It is not necessarily recommended that you too use that specific configuration, but it is the one that this is being built with and tested against. If you use something else (say MikTeX and TeXStudio for example) and are getting weird bugs, then I can't help you. I have used my TeXLive installation with TeXStudio and it works fine, but I have not tested this template with it.

I use this configuration because VS Code is heavily integrated into my workflow as a roboticist. I write code, a lot, so writing my papers in my code editor is what is the most useful for me. Additionally, VS Code has nice Git (source code management) integration which allows for changes to be tracked. This is similar to the "Track Changes" feature in many word processors (namely Microsoft Word) that many users may be familiar with. Using plain text and Git differs in that it will be able to track things like changes to your equation definitions. You won't have to convert back to linear mode, make the change and then re-render. Personally, I do not find the version control aspect a particularly compelling argument for people who do not already know how to and use Git in their workflow. However, it is a nice feature to have if you are already using it.

With that said, I have some additional recommendations for setting up Visual

Studio Code for L^AT_EX editing. Notably, the [LaTeX Workshop extension](#) comes with `chktex` a L^AT_EX linter that will check your document for common mistakes. As anyone who is mildly familiar with L^AT_EX knows, whatever program you're using frequently will spit out a lot of warnings about things despite rendering your document completely correctly as you intended. Underfull hbox anyone? Frankly its one of the things that I dislike about L^AT_EX and why I prefer to use Markdown for most of my writing now. However, L^AT_EX is still the gold standard for typesetting and formatting formal documents, especially in academia and scientific research. So we use it even if it means we keep cursing at it. Eventually something will come along that is both compatible and better.

Anyway, open your settings in VS Code and search for “chktex”. Find the section “Latex-workshop:Linting:Chktex:Exec:Args”. At time of writing this is the fourth item down. I would add the following arguments to the list: `-n2`, `-n8`, `-n12`, `-n24`, and `-n38`. I recommend these because, like markdown, I want to keep the source code text and the rendered product as simple, straightforward, and as similar as possible. If I put a space between items in the code, I would like there to be a space. I don't want to have to explicitly use a tilde.

`-n2` suppress the non-breaking space warning. This frequently comes up with references and citations in text. For example if I cite Doe, 2023 as such, this works and looks correct. In the code this is written as:

```
...\verb|if I cite \cite{doe2023example}|...
```

`chktex` will complain about the space between “cite” and the `\cite` command. I don't really know why but removing that space renders the document as such:

“...if I citeDoe, 2023...”.

This is not what I want. I want the space to be there. To a certain extent I get this, the code is being rendered precisely where you tell it to be rendered in the text,

however L^AT_EX doesn't do well with intentional white space. So I add `-n2` to the list of arguments to suppress that warning and either include or exclude the space between the citation and the text as per the style guide of whatever document I'm writing. For dissertations and theses (and this example) they follow the APA style guide which uses the Name, Year scheme and requires a space between the citation and the text. The bracketed number IEEE style does not require a space.

`-n8` suppresses the warning about potentially incorrect hyphen usages. Some packages (namely the `lipsum` dummy text package) will allow you to specify ranges, lists, bounds et cetera in the bracketed parameters block, e.g.: `lipsum[1-3]`. L^AT_EX however will complain about this as it doesn't know if this is some sort of code parameter, a hyphen between words (e.g.: "co-ordinate"), or an emdash (this thing: —, the fancy long dash). These are pretty distinct in code, particularly the emdash: ---.

`-n12` similarly suppresses interword spacing warnings with '`\` '. Again, if I want a space, I want a space.

`-n24` suppresses the warning spaces in labels and captions for figures and tables. I've never actually seen an issue caused by this space being there. For example in the following snippet:

```
\begin{figure}
  \centering
  \includegraphics[width=0.5\textwidth]{example-image-a}
  \caption{An example figure.}
  \label{fig:example-image-a}
\end{figure}
```

`chktex` will complain about the space (indentation) in front of `label` yet this renders and is captioned and cross-referenced correctly.

-n38 suppresses the warning about punctuation in front of quotes. I have no idea why this is a problem.

Chapter 3 Methods

In this chapter, we'll discuss the specific style guidelines and to use the template formatting and built in \LaTeX features in a way that aligns with the style guide. For starters, you cant take a look at the front matter of this document. The class inherits the `abstract` environment from the `report` class and modifies it a bit to be centered horizontally and vertically on the page. New environments that do the same thing are created for the “Acknowledgments” and “Dedication” sections.

Also take a look at the preamble portion of the the \LaTeX source code. There you will find several self-explanatory commands that set certain variables in the class. These, when combined with the typical `maketitle` command, will generate the title page and copyright page.

Everything else is handled with built in commands. The class definition provides the formatting of sections headers appropriately. The style guide provides for five levels of headings. All headings are numbered (e.g. Chapter 4, Section 4.1) with appendix level 1 headings lettered (e.g. Appendix A). The style guide permits two different heading styles, APA style which is available to every discipline, and “hard sciences” style. For the sake of appealing to the broadest possible group and not making this over hard, the APA style was selected. These headings are accessed via the commands shown in Table 3.1. Individual authors can choose to number or not number subsections using the asteriks verions of the section commands.

Table 3.1: Heading styles and their descriptions. Table captions precede the table.

Level	Command	Description
1	<code>\chapter{}</code>	Chapter heading; Centered, Boldface, title case.
2	<code>\section{}</code>	Section heading; flush left, boldface, title case.
3	<code>\subsection{}</code>	Subsection heading; flush left, boldface italic, title case.
4	<code>\subsubsection{}</code>	Subsubsection heading; indented, boldface, title case.
5	<code>\paragraph{}</code>	Paragraph heading; indented, boldface italic, title case.

3.1 Bibliography and Citations

L^AT_EX is great for managing citation. In the preamble portion of the source code you’ll see the following two lines:

```
\usepackage[style=apa]{biblatex}
\addbibresource{references.bib}
```

The style can be set to any style supported by `biblatex` (e.g. “`ieee`”). The `references.bib` file is a BibTeX file that contains the references for the document. You can add your own references to this file, and L^AT_EX will automatically generate the bibliography for you.

3.2 Block Quotes

Sometimes, you need to quote a very long passage of text. In this case, you can use the `quotation` and `singlespace` environments. This will indent the text and set it in a different font size. For example, the following code:

```
\begin{quotation}
  \begin{singlespace}
    This is a quote from a paper. It should be indented. It should
    be single spaced. It should not have quotation marks. This is
    an example of how to format a very long quote of forty or more
    words in LaTeX.
  \begin{flushright}
    \cite{doe2023example}
```

```

\end{flushright}
\end{singlespace}
\end{quotation}

```

Will produce the following quotation when compiled:

This is a quote from a paper. It should be indented. It should be single spaced. It should not have quotation marks. This is an example of how to format a very long quote of forty or more words in LaTeX.

Doe, 2023

3.3 Figures and Tables

Figures are created using the `figure` environment along with the `includegraphics` command which is included in the class definition which requires the `graphicx` package. Figures, like tables as we’ve seen before are centered on the page and include a caption. Figures are captioned below the image, tables above the table. The following code produces a figure from an image file `example-image-a`. You can point `LaTeX` to a source directory or include the file path. You don’t need to include the file extension.

```

\begin{figure}
\centering
\includegraphics[width=0.5\textwidth]{example-image-a}
\caption{An example figure.}
\label{fig:example-image-a}
\end{figure}

```

Sometimes, and rather notoriously, figures and tables don’t really appear where you want them, or where they are relative to other things in the source code. For instance Table 3.2 is defined after this paragraph in the source code. `LaTeX` will always take care of the typesetting and placement, but you can cajole it to place things like tables, figures, et cetera where you define them by using the `ht` option: `\begin{figure}[ht]` or `\begin{table}[ht]`.

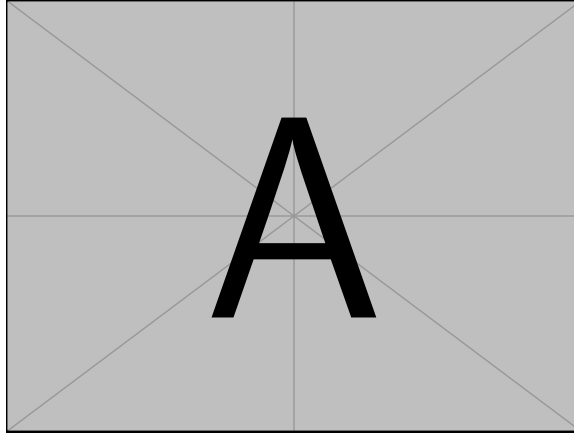


Figure 3.1: An example figure.

A	B	C
1	2	3
4	5	6
7	8	9

Table 3.2: An example table.

3.4 Definitions, Lemmas, and Theorems

Some disciplines, namely mathematics and physics, like to make use of specific formal acknowledgement for things like lemmas, propositions, definitions, notations, and the like. For example something like this:

Notation 3.1 *We can make note of how we use symbols.*

and we can even cross-references them such “Recall Notation 3.1.” These are defined as environments (e.g. `\begin{notation}`) using the `\newtheorem` command:

```

\newtheorem{notation}{Notation}[chapter]
\newtheorem{rem}{Remark}[chapter]
\newtheorem{lem}{Lemma}[chapter]
\newtheorem{cor}{Corollary}[chapter]
\newtheorem{tem}{Theorem}[chapter]
\newtheorem{prop}{Proposition}[chapter]
\newtheorem{example}{Example}[chapter]
\newtheorem{define}{Definition}[chapter]

```

This will create a new environment for each of these items. The `[chapter]` option will number them according to the chapter they are in. For example, the first lemma in Chapter 2 will be numbered “Lemma 2.1” and the second will be “Lemma 2.2”. The first lemma in Chapter 3 will be “Lemma 3.1” and so on.

3.5 Equations

Equations are created using the `equation` environment. This will automatically number the equation for you. For example, the following code:

```
\begin{equation}
  E = mc^2
  \label{eq:example}
\end{equation}
```

Will produce the following equation when compiled:

$$E = mc^2 \tag{3.1}$$

This will automatically number the equation for you. You can reference the equation using the `\ref` command. For example, Equation 3.1 is an example of an equation.

3.6 Conclusions

Other than the specific environment instances noted in this chapter, everything else is automatically formatted for you. This formatting is demonstrated in the the next chapter.

Chapter 4 Headings Examples

4.1 Section: This Heading Should be Flush Left, Boldface, and Title Case.

This is a section heading. It should be flush left, boldface, and title case. The text should be indented at the start of a new paragraph, but not for the first paragraph in the section. The class template automatically takes care of the title case for you. The following words are excluded based on APA suggestions.

- **Articles:** a, an, the
- **Prepositions:** as, at, an, by, for, in, of, off, on, per, to, up, via
- **Conjunctions:** and, as, but, for, if, nor, or, so, yet
- **“Be” verbs:** be, am, is, are, was

Subsubsection: Indented, Boldface, Title Case Heading Ending With a Period. Paragraph text continues on the same line of the paragraph. Period should be included in the subsection title within the curly brackets.

Subsubsubsection: Indented, Italic, Title Case Heading Ending With a Period. The paragraph text continues on the same line as the heading. This heading is access via the `\paragraph` command. Again, the period should explicitly be included in the title contained within the curly brackets.

Bibliography

Doe, J. (2023). *An example book*. Fictional Publishing House.

Appendix A Appendix Formatting

Appendices are for including additional material that is not essential to the main body of the document, but is still relevant. This can include things like additional figures, tables, or code snippets. Appendices are numbered with letters (e.g. A, B, C, etc.) and are formatted like chapters. The `\appendix` command will automatically change the chapter numbering to letters. To change the chapter names from “chapter” to “appendix” use `\renewcommand{\chaptername}{Appendix}`. This line is included in the source template file right after the `\appendix` command.

A.1 Appendix Section

Appendices can have sections, subsections, and subsubsections just like the main body of the document. The only difference is that the chapter number is replaced with a letter (e.g. A, B, C, etc.). The section numbers are still numbered as they are in the main body of the document (e.g. A.1, A.2, etc.).

A.1.1 Appendix Subsection

Appendix subsections are numbered as they are in the main body of the document (e.g. A.1.1, A.1.2, etc.).

Appendix Subsubsection. Appendix subsubsections are like this.

Appendix Subsubsubsection. Appendix susubsubsections again use the paragraph command.