**\LaTeX** Short Course

1pm - 4pm

Tuesday (Feb. 21) & Thursday (Feb. 23)

ME 10a

---

**Basic \LaTeX \textit{2}lex Skills**

R W Kaszeta

(with modifications/updates by Blake Larson)

---

**Schedule**

**Day 1: Basic Skills**

1. Basic \LaTeX Introduction.
2. Editors on MEnet Machines.
3. Running \LaTeX, Previewing, and Printing.
4. Basic \LaTeX Classes and Environments.
5. Math.

---

**Day 2: Intermediate Skills**

1. Large Documents.
2. Figures and Tables.
4. The thesis-me class
5. Add-ins, helpers, and other setups

---

**Advantages of \LaTeX:**

- \LaTeX provides an easy-to-use method for producing high-quality typeset math.
- \LaTeX provides very high quality typeset output which pays attention to fine typographical details such as math typesetting, ligatures (i.e. “fi” instead of “ff”), advanced font handling (with both slanted and italic text, which are slightly different), and hyphenation that are often handled poorly in WYSIWYG programs.
- The \LaTeX markup language lends itself to writing well-structured documents that are easy to modify.
- The user only needs to learn a few easy commands, and generally does not need to tinker with the actual layout of the document.
- Even complicated structures such as footnotes, cross-references,
Disadvantages of \LaTeX:

- \LaTeX is not WYSIWYG—you have to run \texttt{latex} on your file before you can see your changes.
- If you are trying to produce a document for which there is no predefined layout, it requires a fair bit of knowledge to design a new layout.
- You cannot easily exchange \LaTeX files with colleagues that are unfamiliar with it.

Where to find more info:

- \textit{The \LaTeX Book} by Leslie Lamport.
- \textit{The \LaTeX Companion} by Goossens, Mittelbach, and Samarin.
- \textit{The \LaTeX Graphics Companion} by Goossens, Rahtz, and Mittelbach.
- The MEnet online \TeX Docs,
  \url{http://www.menet.umn.edu/docs/tex}
  - The \textit{MEnet FAQ}
  - \textit{Essential \LaTeX++}
  - \textit{The Not So Short Introduction to \LaTeX2ε}

Getting Started:
To get started, we'll first copy all the course materials to a directory on your account:

1. cd ~
2. mkdir latexcourse

Open your favorite web browser to
\url{http://www.menet.umn.edu/~blake/latexcourse/}
Download all of those files into your \texttt{latexcourse} directory

The Official \LaTeX Process
1. Create the text file using an editor, with an ending of \texttt{.tex}.
2. Run \texttt{latex} on the \texttt{.tex} file to format the text and create a \texttt{.dvi} file.
3. Print or preview the \texttt{.dvi} file.

Our Approach for this Course
1. Create the text file as before, with a \texttt{.tex} extension
2. Run \texttt{pdflatex} on the \texttt{.tex} file to create a standard \texttt{.pdf} file
3. View the \texttt{.pdf} file using \texttt{acroread}

This approach produces documents with which we are more comfortable working.

Editors:
There are three basic editors that work well with \LaTeX:
1. \texttt{pico}
2. \texttt{emacs}
3. \texttt{vi}
4. \texttt{nedit}
5. Notepad/Wordpad (Windows)
6. TeXnicCenter or other Windows-based \LaTeX editors
Our First Document

1. Open the sample file example1.tex in your favorite editor
2. Note the basic parts:
   \documentclass declaration
   \begin{document}
   \author, \title, and \maketitle
   — document contents —
   \end{document}
3. Run pdfLaTeX example1
4. Run \texttt{acroread example1.pdf}

Document class options

- Can change font size, etc., in most classes:
  \documentclass[11pt,a4paper]{article}

Exercise: Change the document class and try different font sizes for our example document

Running Text:

1. Lines and Paragraphs
   - You can force new lines with \textbackslash\textbackslash.
   - You can force new pages with \texttt{\textbackslash page}.
2. Hyphenation
   - You can enforce a hyphenation pattern with \texttt{\textbackslash hyphenation{}}.
   - You can keep hyphenation from occurring with \texttt{\textbackslash mbox{}}.
3. Quotes and Dashes
   - Quotes are entered with \texttt{‘}, \texttt{"}, \texttt{‘}, and \texttt{‘} (not \texttt{‘})
   - Dashes are entered with \texttt{--}, \texttt{---}, and \texttt{---}.

Symbols:

\begin{verbatim}
\dag \ddag \S \copyright
\oe \OE \aa \AE
\& \O \L \E
\textbackslash idots \LaTeX\TeX
\end{verbatim}
Sectioning Commands:
\part\%
\chapter\%
\section\%
\subsection\%
\paragraph\%
\subparagraph%

And each has a “starred” form (i.e. \section*{}).

Title Pages:
To make a title page, you first have to tell \LaTeX{} about the paper:
\author{}
\date{}
\title{}

and then create the title page with \maketitle.

Labels and References:
At almost any point in the document you can place a “label” using \label{key}, where the argument is a “key”, a short one work description of the location.

You can refer to this label by section and page number using \ref{key} and \pageref{key}, respectively.

Advanced: you can use \autoref{key} to automatically place the proper label (i.e. Section 2.1 or Table 1.1 instead of just 2.1 or 1.1).

Must include the hyperref package.

Other Typefaces:

<table>
<thead>
<tr>
<th>Command</th>
<th>Declaration</th>
<th>Effect</th>
</tr>
</thead>
<tbody>
<tr>
<td>\ttfamily...</td>
<td>\ttfamily...</td>
<td>Text is set in typewriter family</td>
</tr>
<tr>
<td>\sffamily...</td>
<td>\sffamily...</td>
<td>Text is set in serif family</td>
</tr>
<tr>
<td>\textfamily...</td>
<td>\textfamily...</td>
<td>Text is set in roman family</td>
</tr>
<tr>
<td>\mdseries...</td>
<td>\mdseries...</td>
<td>Text is set in medium series</td>
</tr>
<tr>
<td>\upshape...</td>
<td>\upshape...</td>
<td>Text is set in upright shape</td>
</tr>
<tr>
<td>\textupshape...</td>
<td>\textupshape...</td>
<td>Text is set in standard shape</td>
</tr>
<tr>
<td>\textslshape...</td>
<td>\textslshape...</td>
<td>Text is set in slanted shape</td>
</tr>
<tr>
<td>\textscshape...</td>
<td>\textscshape...</td>
<td>Text is set in small caps shape</td>
</tr>
<tr>
<td>\textem...</td>
<td>\textem...</td>
<td>Text is set emphasized</td>
</tr>
<tr>
<td>\textnormal...</td>
<td>\textnormal...</td>
<td>Text is set in document font</td>
</tr>
</tbody>
</table>

Sizes:
\tiny \scriptsize \footnotesize \small \normalsize \large \Large \LARGE \huge \Huge

Note these are declarations, so the proper syntax is \Large \text{Text}.

Environments:
For special purpose text such as lists, quotations, poetry verse, and aligned text, \LaTeX{} provides environments.

To use an environment, you use \begin{} and \end{}:
\begin{environment}
\end{environment}

where environment is the name of the particular environment.
Lists:
Two of the simplest environments are list environments. The first is the \texttt{itemize} environment. Each item starts with \texttt{item}:
\begin{itemize}
\item First item.
\item Another item.
\item Yet another item.
\end{itemize}
The \texttt{enumerate} environment does the same thing, but numbers the entries.

Exercises:
- Try to typeset a list some accented words in \textbf{bold}, \textit{italics}, and various font sizes.
- Make a few different sections and label them. Use cross-references to refer to these sections. What happens when you run \texttt{pdflatex} on the command line?

Math:
\LaTeX{} provides two basic methods of typesetting math, “inline” which occurs inside of a sentence: \( a = \frac{b}{c} \), and “displayed”, which occurs centered between paragraphs:
\[
\frac{b}{c}
\]
To typeset material in inline mode, surround it with dollar signs: \$...\$. Thus, \$A=\frac{b}{c}\$ yields \( A = \frac{b}{c} \).

Displayed Equations
To type displayed math, use \texttt{\begin{align} and \texttt{\end{align}} instead. So \texttt{\begin{align}A &= bh\end{align}} yields
\[
A = bh
\]
If instead you use the \texttt{equation} environment instead of \texttt{\begin{align} and \end{align}}, you get the same results, but with an equation number:
\[
A = bh \tag{1}
\]
Note that if you put a \texttt{\label} in here, you can refer to that equation number elsewhere in the text.

Differences from Text Mode:
- Spaces in math mode are ignored.
- No empty lines are allowed.
- Letters are italicizes
- All letters are typed as single variables, not words. So \$\texttt{effective}\$ (effective) typesets much differently from “effective”
Basic Math Building Blocks

Sub/superscripts are produced with _ and ^, For example, \( p_2 \) gives \( p_2 \) and \( x^y \) gives \( x^y \). Note these only work on a single character—to sub- or superscript an expression surround it with curly braces.

Fractions are produced using \( \frac { } { } \), which takes two arguments, one for the numerator and one for the denominator.
\[ \frac{a}{b+c} \] yields \( \frac{a}{b+c} \).

Radicals are produced using \( \sqrt { } \), with \( \sqrt{x} \) yielding \( \sqrt{x} \) and \( \sqrt[n]{x} \) yielding \( \sqrt[n]{x} \).

Lowercase Greek Letters are given by simply spelling out the letter as a macro, \( \alpha \) yielding \( \alpha \), etc.

Uppercase Greek Letters are the same, except the macro name is capitalized, \( \Omega \) yielding \( \Omega \), etc.

Math Mode Accents Math mode has its own set of accents, similar to the ones used in text mode:
\[ \hat{a} \] \( \hat{a} \)
\[ \acute{a} \] \( \acute{a} \)
\[ \dot{a} \] \( \dot{a} \)
\[ \breve{a} \] \( \breve{a} \)
\[ \check{a} \] \( \check{a} \)
\[ \grave{a} \] \( \grave{a} \)
\[ \vec{a} \] \( \vec{a} \)
\[ \dot{a} \] \( \dot{a} \)
\[ \tilde{a} \] \( \tilde{a} \)

Exercises
How would you typeset:
1. \( y = \sqrt{2x_2 + w_2} \)
2. \( \alpha \)
3. \( n^{1/3} \)
4. \( \tau_{xy} \)

Answers
How would you typeset:
1. \( y = \sqrt{2x_2 + w_2} \) \( \Rightarrow \) \( y = \sqrt{2x_2 + w_2} \)
2. \( \alpha \) \( \Rightarrow \) \( \alpha \)
3. \( n^{1/3} \) \( \Rightarrow \) \( n^{1/3} \)
4. \( \tau_{xy} \) \( \Rightarrow \) \( \tau_{xy} \)

Spacing
Sometimes you may wish to “tweak” the spacing in math mode. The following macros are provided for this purpose:
\( \ \) thin space \( \ \) medium space
\( \backslash \) negative thin space \( \backslash \ ) thick space
Integrals and Summations

To do integrals and summations, simply put in a $\texttt{\textbackslash int}$. For example, $\texttt{\textbackslash int x \textbackslash dx}$ (note the spacing factor) yields

$$\int x \text{\,} \text{\,} dx$$

To add limits, simply sub and superscript the integral, $\texttt{\textbackslash int}_0^\infty \text{\,} x \text{\,} \text{\textbackslash dx}$,

$$\int_0^\infty x \text{\,} \text{\,} dx$$

To do summations, simply do the same thing with $\texttt{\textbackslash sum}$.

Parentheses, Braces, and other delimiters

To use parentheses and braces, simply type \{$ and \}$ for curly braces, you have to type \{ and \} for defining groups.

However, if you want to type large expressions containing symbols such as integrals and fractions, it is nice to have larger versions of these delimiters. If you preface the delimiters with \texttt{\textbackslash left} and \texttt{\textbackslash right}, you get larger symbols that are (usually) the right size.

Example:

$$\texttt{\textbackslash left} \text{\,} \frac{\text{\textbackslash int}_0^1 \text{\,} \text{\textbackslash dx}}{\text{\textbackslash Delta \, x}} \text{\,} \texttt{\textbackslash right}$$

Changing Font Faces in Math Mode

Like text mode, you can change the typeface used in math mode. The appropriate commands are \texttt{\textbackslash mathrm}, \texttt{\textbackslash mathit}, \texttt{\textbackslash mathbf}, \texttt{\textbackslash mathsf}, and \texttt{\textbackslash mathtt} which all work similarly to the text versions of these commands.

Note that if you have variables or subscripts that are more than a single character long, you should process them with \texttt{\textbackslash mathit} to get the correct spacing.

Exercises

Typeset the following:

1. $1 + \left(\frac{1}{1-p}\right)^3$
2. $\pi(n) = \sum_{k=2}^{n} \left[ \frac{\phi(k)}{k-1} \right]$
3. $\Delta x = x_{\text{max}} - x_{\text{min}}$
4. $A = \int_0^\infty r^2 \text{\,} \text{\,} dr$

Answers

Typeset the following:

1. $1 + \left(\frac{1}{1-p}\right)^3$
2. $\pi(n) = \sum_{k=2}^{n} \left[ \frac{\phi(k)}{k-1} \right]$
3. $\Delta x = x_{\text{max}} - x_{\text{min}}$
4. $A = \int_0^\infty r^2 \text{\,} \text{\,} dr$

Common Mistakes

Sometimes \LaTeX{} will give you errors when you try to run it because of problems with your input file. As mentioned earlier, sometimes the location of the error that \LaTeX{} reports may not be correct, so it is useful to know what usually causes these problems. Usual sources of error include:

- Misspelled command or environment names
- Missing or improperly nested \texttt{\textbackslash end} statements.
- Improperly matched \{ and \}. They should always come in pairs.
- Missing command arguments.
- A missing $\&$
- Using one of the special \LaTeX{} characters such as #.
Macros

The simplest macro is a substitution macro, where it substitutes a given string of \LaTeX\ commands in place of the command.

For example, if your document contains the string “Taylor-Görtler” a number of times, we can define the command \texttt{\textbackslash TG} as an abbreviation:

\begin{verbatim}
\newcommand{\TG}{Taylor-G"ortler}
\end{verbatim}

And now you can simply type \texttt{\TG} instead of Taylor-Görtler.

Exercise:

Write a macro that prints your full name \textit{in boldface}.

\begin{verbatim}
\newcommand{\RWS}{\textbf{Richard W. Kasza\texttt{\textbackslash e}}} \newcommand{\hatv}{\hat{\textbf{v}}}
\end{verbatim}

This method also works in math mode. For example, let’s say your document uses the expression $\hat{v}''$, which is typed as $\hat{\textbf{v}}''$, which is not something you want to have to type repeatedly. So we can define \texttt{\textbackslash html} as shorthand for this expression by saying

\begin{verbatim}
\newcommand{\hatv}{\hat{\textbf{v}}} \newcommand{\tau}{\textbf{\textbackslash t}}
\end{verbatim}

So now instead of typing $\hat{\textbf{v}}''$, you can instead simply type $\hatv''$.

Macros can also take \textit{arguments}, just like some of the built-in functions. For example, if you are typesetting partial derivatives ($\frac{\partial f}{\partial x}$), the \LaTeX\ code generally would look like:

\begin{verbatim}
\frac{\partial f}{\partial x}
\end{verbatim}

Note that every time you enter a derivative it is of the same form except for portions of the numerator and denominator. Thus, you could define a macro that takes two arguments, the numerator and denominator without the $\partial$

\begin{verbatim}
\newcommand{\ppfrac}[2]{\frac{\texttt{#1\textbackslash t\texttt{\textbackslash e}}\{\texttt{#2\textbackslash t}\}}{\texttt{#1\textbackslash t\texttt{\textbackslash e}}\{\texttt{#2\textbackslash t}\}}}
\end{verbatim}

Where the \texttt{#1} and \texttt{#2} are the first and second arguments. So you could then produce $\frac{\partial f}{\partial x}$ with $\ppfrac{x}{y}\texttt{\partial x}$.
Exercise:
Produce a macro that typesets
\[
\frac{\partial A}{\partial x} \Delta x + \frac{\partial A}{\partial y} \Delta y
\]
where the $A$, $x$, and $y$ are arguments 1, 2, and 3, respectively.

Answer:
Produce a macro that typesets
\[
\frac{\partial A}{\partial x} \Delta x + \frac{\partial A}{\partial y} \Delta y
\]
where the $A$, $x$, and $y$ are arguments 1, 2, and 3, respectively.
\[\text{\texttt{\textbackslash{}newcommand\{\texttt{series}\}\[3\]}}\]
\[\text{\texttt{\textbackslash{}frac\{\texttt{partial} 1\}\{\texttt{partial} 2\}\Delta 2}}\]
\[+\text{\texttt{\textbackslash{}frac\{\texttt{partial} 1\}\{\texttt{partial} 3\}\Delta 3}}\]

Examples:
You can even do some fairly complicated things with macros. For example, this macro provides “prescripts” (subscripts and superscripts preceding an expression instead of following it):
\[\text{\texttt{\textbackslash{}newcommand\{\texttt{prescript}\}\[2\]}}\texttt{\textbackslash{}ensuremath\{\texttt{bx}\}_\{\texttt{#1}\}^{\texttt{#2}}\}\]
So that \texttt{\textbackslash{}prescript(x,y)A\}} produces
\[
\prescript{y}{x}{A}
\]

User-defined Environments:
There is also a \texttt{newenvironment} command for defining your own environments. The first argument is the new environment name, the second argument is the commands you want to occur at the beginning of the environment, and the third argument is the commands you want to occur at the end of the environment.
For example, if you want to create your own quotation environment which is typeset in italics, you’d enter
\[\text{\texttt{\textbackslash{}newenvironment\{myquotation\}}}}\]
\[\text{\texttt{\textbackslash{}begin\{quotation\}}}}\text{\texttt{\textbackslash{}em\}}\text{\texttt{\textbackslash{}end\{quotation\}}}}\]

Exercise:
Produce an environment that typesets a list in boldface.
\[\text{\texttt{\textbackslash{}newenvironment\{bflist\}}}}\]
\[\text{\texttt{\textbackslash{}begin\{itemize\}}}}\text{\texttt{\textbackslash{}bfseries\}}\text{\texttt{\textbackslash{}end\{itemize\}}}}\]
Large Documents
When processing a large document such as a thesis, it is often desirable to deal with a number of small files instead of one large file. To accomplish this, \LaTeX provides the `\include` command, `\include{chap1}` which will insert the contents of `chap1.tex` into the current document. Thus, you could create a multichapter document like so:

```
\documentclass[book]
\begin{document}
  \include{chap1}
  \include{chap2}
  \include{chap3}
  \include{chap4}
  \appendix
  \include{app1}
  \include{app2}
\end{document}
```

Using subfile
The `subfile` package is another convenient way work with multiple chapters or sections within a larger document. Its usage is similar to `\include`:

```
\documentclass[letterpaper,12pt]{thesis-mc}
\usepackage{subfiles}
\mainmatter
\subfile{ch-intro.tex}
\subfile{ch-concl.tex}
\appendix
\subfile{appendix.tex}
\end{document}
```

Tables
The `tabular` environment takes a single argument, which tells it how to align each column. This argument is a series of letters, one for each column, that tells how that column is formatted: `c` for centered, `l` for left, and `r` for right. The environment itself consists of a series of rows, separated by `\`. In each row, each column’s material is separated by `&`.

Thus, to create a table with “a” and “b” in the top row, and “c” and “d” in the bottom row all aligned left, the code would be:

```
\begin{tabular}{ll}
a & b \\ c & d \\
\end{tabular}
```
which produces
```
<table>
<thead>
<tr>
<th>a</th>
<th>b</th>
</tr>
</thead>
<tbody>
<tr>
<td>c</td>
<td>d</td>
</tr>
</tbody>
</table>
```

Additionally, one can add vertical lines between the columns by adding a `|` to the `tabular` mode argument:

```
\begin{tabular}{|ll|}
  one & two \\
  three & four \\
\end{tabular}
```
produces
```
<table>
<thead>
<tr>
<th>one</th>
<th>two</th>
</tr>
</thead>
<tbody>
<tr>
<td>three</td>
<td>four</td>
</tr>
</tbody>
</table>
```
Horizontal lines are added by placing a `\hline` after the `\`. A line across only columns \( i \) through \( j \) is done with

\[ \textbackslash \text{line} (i-j) \]

So, this table,

\begin{tabular}{c|c|c}
one & two \& three & \\hline
1 \& 2 \& 3 \& \\hline
\end{tabular}

produces

| \hline one \& two \& three
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>1 &amp; 2 &amp; 3</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

One can also create material that spans multiple columns (like table headers) using `\multicolumn`. The usage is

\[ \textbackslash \text{multicolumn} \{number \text{ of \ columns}\} \{alignment\} \{material\} \]

As an example,

\begin{tabular}{c|c|c|c|c|c}
\hline
& a \& b \& & & \\
\hline
\end{tabular}

produces

| \hline a \& b

Arrays:
\LaTeX also provides the `\texttt{array}` environment, which is almost identical to `\texttt{tabular}` mode except that it formats its contents in math mode.

\[ A = \begin{bmatrix} x_1 & x_2 \\ x_3 & x_4 \end{bmatrix} \]

produces

\[
A = \begin{bmatrix} x_1 & x_2 \\ x_3 & x_4 \end{bmatrix}
\]

Basic Graphics Handling:
\LaTeX provides the ability to include *encapsulated postscript* files as graphics in your document through the `\texttt{graphicx}` package. Note that most graphics programs (for example, Matlab, Gnuplot, and Photoshop) can produce encapsulated postscript as a standard option. And if your program does not produce encapsulated postscript, you can use `\texttt{xy}` on MEnet Unix workstations to convert your file to `\texttt{eps}` format.

To insert a PostScript file into your document, you must first load the `\texttt{graphicx}` package using

\[
\texttt{usepackage{graphicx}}
\]

XFig Graphics
To include \texttt{XFig} graphics, the steps are

1. Make the drawing in \texttt{XFig}.
2. Make sure any text you want formatted by \LaTeX is marked as “special” in the “text properties” box.
3. “Export” the drawing as “Two Part \LaTeX/Postscript (postscript part)” in portrait mode.
4. “Export” the drawing as “Two Part \LaTeX/Postscript (LaTeX Part)”.
5. Make sure your document includes the `\texttt{graphics}` package.
6. Load the drawing into the document using

\[
\texttt{input(file.ps \texttt{t})}
\]
**Figures and Tables:**

\LaTeX{} provides a useful mechanism for displaying figures and tables—a “float”. If you insert a \texttt{figure} or \texttt{table} environment in your document, \LaTeX{} will allow the material in that environment to “float” to a convenient location in the document (generally, the current location, or the top of the next page if that would fit better).

Additionally, you can label a figure or table with the \texttt{\caption{}} command, which creates a numbered caption for the environment. The commands \texttt{\listoffigures} and \texttt{\listoftables} can then be used in the document to create a list of figures or a list of tables, respectively. And if you place a caption on a drawing, you can use \texttt{\label{}} and \texttt{\ref{}} to create cross-references to that figure.

---

**Bibliographies:**

One of the most powerful features of \LaTeX{} when used with large papers such as theses is its handling of bibliographies. \LaTeX{} provides two methods for creating bibliographies, \texttt{manual} and \texttt{automatic}.

---

**Example:**

\begin{thebibliography}{99}
\item \texttt{\bibitem{kaszeta} R. W. Kaszeta, \texttt{Intermediate \LaTeX\} \& on MEnet Systems}, 1997
\end{thebibliography}

produces

**References**


---

**Manual Bibliographies:**

The first step is to create your bibliography list. This is accomplished by creating a \texttt{thebibliography} environment. This environment takes a single argument which is a piece of text the same width as the widest item label in the list. This environment is like an \texttt{enumerate} environment except that each item is \begin{itemize} with a \begin{itemitem} command whose argument is a “citation key” (a text string that is unique to that reference).

---

**Citations:**

You can then create references to each bibliography entry using \texttt{\cite{key}}. For example, you can refer to the reference above by inserting \texttt{\cite{kaszeta}} to make a reference like [1].
**BibTeX:**
A more advanced (but admittedly more complicated) way of creating bibliographies is to use BibTeX. BibTeX takes a list of all the citation keys used in a document, extracts bibliography information from a database (called a `.bib` file), and formats it (according to style rules in a `.bst` file) for inclusion in your document.

The first step is to include the `natbib` package which provides support for citations in your text.

Next, you place a `\bibliography` command in your document where you wish the bibliography to appear. The argument of this command is a comma-separated list of `.bib` files containing the bibliographical database for the document.

Additionally, you must insert a `\bibliographystyle` command in the document preamble which tells BibTeX which bibliography style file (.bst) to use. For most purposes, I recommend

```
\bibliographystyle{menet}
```

Then, you simply insert `\cite` commands in your document where you wish to create citations in your document, using the keys from the `.bib` file.

Once you have done this, you generate the bibliography list using the following steps:

1. Run `BibTeX` once, which creates a list of citation keys for `BibTeX` to find information for.
2. Run `BibTeX` by typing `bibtex myfile` to generate a list of bibliography entries.
3. Run `BibTeX` to insert the results of the `BibTeX` run.
4. Run `BibTeX` once more to ensure the cross references are correct.

**The thesis-me Class**
This class file provides a modified “book” class for `LaTeX2e`, so most anything that would work for a “book” will work in your thesis. The changes from the standard “book” class are:

- The default paragraph indent is set to 0 points.
- The default paragraph separation is set to 10 points.
- The page margins are set at 1.5 inches on the left, 1 inches elsewhere.
- The linespacing is set to 1.5.
- The section, subsection, and subsubsection headers have been changed so they aren’t as large as the `LaTeX2e` defaults.

**What this class doesn’t do**
This package only makes the changes required by the Graduate School, it doesn’t make changes for user preferences. Since the package is basically compatible with the standard `LaTeX2e` classes, the standard `LaTeX2e` customization methods and packages can be used.

- The `\maketitle` macro has been changed to produce a signature page, a title page, and a copyright page of the correct format.
- Page numbering is set so that the title pages are unnumbered, front matter is numbered with roman numerals, and all subsequent pages are numbered with arabic numerals.
- Two-sided typesetting is disabled.
- `abstract`, `dedication`, and `acknowledgments` environments are defined for use in the front matter.
- The `encnotes` document class option redefines the `\footnote` macros to place footnotes at the end of the chapter instead of the bottom of the page.
Using this class:
The first step to using this class file is to tell \texttt{\LaTeX} that you want to use the \texttt{thesis-me} class:
\begin{verbatim}
\documentclass[\thesis-me]
\end{verbatim}
which loads the class file.
Then, you must provide the following information:
\begin{verbatim}
\Author{Author Name}
\Title{Thesis Title}
\Month{Month of Thesis Publication}
\Year{Year of Thesis Publication}
\Adviser{Adviser Name}
\end{verbatim}
and if you are a Masters student you should also insert
\begin{verbatim}
\Degree{MASTER OF SCIENCE}
\degreelevel{masters}
\end{verbatim}

Main Matter:
After the front matter you are ready to begin the main document, so you enter a \texttt{mainmatter} before the first chapter to tell \texttt{\LaTeX} to again reset the page counter and start numbering pages with arabic numerals.

After all the chapters, you enter the macro \texttt{appendix} before the appendix chapters so that \texttt{chapter} will start labelling the chapters as appendices.

The \texttt{backmatter} macro can be used after the appendices for further chapters that you don't want titled as "chapters", such as an index.

Endnotes:
If you would like to use endnotes instead of footnotes, the first step is to tell the document class you want endnotes:
\begin{verbatim}
\documentclass[endnotes]{thesis-me}
\end{verbatim}
Then instead of using \texttt{footnote}, you use \texttt{endnote} instead. At the end of each chapter (or wherever you want the endnotes to appear) you can list all of the unprinted endnotes using \texttt{\showendnotes}.

Front Matter:
After the title pages, you may wish to include an abstract and dedication; these are done with the \texttt{abstract} and \texttt{dedication} environments.

All "front matter" (everything after the title page and before the first chapter) should be preceded by the macro \texttt{frontmatter} which resets the page counter and tells \texttt{\LaTeX} to number the pages with roman numerals.

Useful things to include in the front matter include
\begin{verbatim}
\tableofcontents
\listoffigures
\listoftables
\end{verbatim}

Line Spacing:
This class defaults to 1.5 spacing for the document. If you would like to doublespace the thesis instead, simply insert a \texttt{\doublespacing} at the beginning of your document.

If you have environments like quotations, or entire listings like your bibliography which you would like singlespaced, you can include them inside a \texttt{\singlespace} environment.

User Configuration:
In order to allow simple user configuration of this package without needing to modify \texttt{thesis-me.cls}, the class file loads in the tex file \texttt{thesis.cfg} (stuck someplace where your \texttt{TEXINPUTS} setting will find it) after it has done all of its configuration. This is a useful place to override the \texttt{thesisme} default settings and provide macros you always wish to have handy when typesetting theses.
Other Useful Macro Packages:
There are a number of additional macro packages that are useful for
typesetting your thesis. All of these are documented on the
MEnet TeX documentation page.

<table>
<thead>
<tr>
<th>Package</th>
<th>Function</th>
</tr>
</thead>
<tbody>
<tr>
<td>graphic</td>
<td>Inclusion of graphics</td>
</tr>
<tr>
<td>rotating</td>
<td>Allows landscape tables and figures.</td>
</tr>
<tr>
<td>nomencl</td>
<td>Produces nomenclature lists.</td>
</tr>
<tr>
<td>natbib</td>
<td>Better handling of bibliographies.</td>
</tr>
</tbody>
</table>

Making Slides:
The easiest way of creating slides and overheads with \TeX is to
use the \texttt{seminar} document class:
\begin{verbatim}
\documentclass{seminar}
\end{verbatim}

To create each slide, you simply insert a \texttt{slide} environment:
\begin{verbatim}
\begin{slide}
\SlideMaterial
\end{slide}
\end{verbatim}

The \texttt{seminar} document class automatically increases the type size
to be easier to read, and places a border around the slide.

Making the slides:
In general, almost all of the environment available in standard
\TeX \texttt{2e} are also available with the \texttt{seminar} package. However,
note that sectioning commands such as \texttt{section} and such aren't
terribly useful.

However, when doing things such as tables and figures it is
important to note that \texttt{seminar} is magnifying the output, so units
such as "4.5in" will appear larger.

Lecture Notes:
Any text inserted in the document that is not in a \texttt{slide}
environment is typeset as lecture notes for the previous slide. Thus,
to annotate a slide in your notes:
\begin{verbatim}
\begin{slide}
\SlideContents
\end{slide}
\end{verbatim}
\texttt{Notes on Slide Contents}

Customization of the \texttt{seminar} class:
A number of simple customizations are possible with the \texttt{seminar}
class:

- If you include the \texttt{fancybox} class option, you can change the
  slide outline to one of "shadow", "double", "oval", or "Oval"
  with the \texttt{\slideframe} command.
- You can also use the \texttt{slide} environment for printing slides in
  "portrait" format. (Be careful when printing).
- You can tell \texttt{seminar} to print just the slides with the
  \texttt{slideonly} package option.

Printing Slides:
Note that the \texttt{seminar} package produces landscape slides by
default. Thus, you should tell dvips the slides are landscape when
printing them:
\texttt{dvips \-P color155 \-t landscape file.dvi}
For more info

Many more customizations are possible, including color, overlays, and fancy frames. For more information consult either:

- The \texttt{seminar} documentation, http://www.menet.umn.edu/docs/tex.
- \texttt{The \LaTeX\ Graphics Companion}.

More Graphics

Previously, we discussed two methods of including graphics in \LaTeX\ 2e documents. Two more methods of graphics inclusion that may be useful will now be discussed:

- The \texttt{psfrag} macro package.
- The \texttt{pstricks} macro package.

The \texttt{psfrag} macro package

The \texttt{psfrag} macro package allows you to replace text strings in \texttt{eps} drawings with fragments of \LaTeX\ code when you include the image. This is especially useful for programs that produce \texttt{eps} files of poor quality in which you’d like to improve the text labels (for example, MATLAB).

Usage

First you must load the package (in addition to \texttt{graphicx}):

\begin{verbatim}
\usepackage{psfrag}
\end{verbatim}

Then, immediately \texttt{before} you include the graphics with \texttt{\usepackage} you enter a \texttt{\psfrag} command to tell \LaTeX\ 2e what text to replace:

\begin{verbatim}
\psfrag{original text}{replacement code}
\end{verbatim}

Note that the “original text” must be the \textit{entire} string and not just part of it.

Also note that to preview the results you must use ghostview instead of xdvi.

For example, here is some \texttt{pstricks} code that produces a diagram of inlet pipe flow:

\begin{verbatim}
\begin{picture}(4.5,1.25)
\psline(0,0)(4.5,1.25)
\psline(0,0)(4.5,0)
\psline(0,0)(0,0,625,0,625,1)
\psline(3.5,0)(3.5,1)
\psline(0,0,0,0.25)(0.625,0.25)
\psline(0,0)(0,0.5)(0.625,0.5)
\psline(0,0.75)(0.625,0.75)
\pcurve(3.5,0)(4.5,0.5)(4.5,0.5)
\pcurve(3.5,0.25)(4.5,0.25)
\pcurve(3.5,0.5)(4.5,0.5)
\pcurve(4.5,0.75)(4.5,0.75)
\psline(3.5,0.75)(4.5,0.75)
\pcurve[linestyle=dotted](3,0)(1.75,0.75)(3,0.5)
\rput[2](1.5,0.5){\footnotesize Core flow}
\rput[2](1.5,0.5){\footnotesize Boundary layer}
\end{picture}
\end{verbatim}
This sample code produces the following output:
\[ \text{[scale=0.6]pipeeps} \]

For more information on the PS-Tricks package, see the MENet \TeX\ documentation page.

- Use LyX for a more graphical implementation of \LaTeX\n- Use MikTeX for Windows platforms
- Consider TeXnicCenter or other \LaTeX\-ready editors for Windows platforms
- Make easy tables using Excel2LaTeX\xds
- Use publisher-provided classes, e.g., ASME, IEEE, Elsevier for conference and journal articles
- \TeXPoint is an add-in for PowerPoint available on the MENET Windows PCs