

3014 total words

Todo list

Find a good example	4
This comment is much too long for the correction margin and would not be displayed legibly	4
Figure: An illustrative image is added here	4
Figure: Here another picture is missing.	5
[CB₁]: Test1	5
[CB₂]: Test3	5
[CB₃]: Test3	5
Figure: [CB₄]: Test4	5
Figure: Missing class diagram.	9

Agent Approach Support
Work Instruction Language Overriding Execute Kotlin
Call Profiling Refactoring Query Context Base Plug-In
Implementation Diff Tool
Snapshot Code Editor
Inheritance Benchmark Execution
Debugging Detection Requirement
Trace Transpilation Annotation Compiler IDE
Delegation Java Interpretation Domain-Specific Language
Check Eclipse Compilation
Scala Object
Operand Bytecode Probe Node
Metamodel Program Type Evaluation Performance
Energy Consumption Analyse Measurement Virtual Machine
Instrumentation Class Python View State Model Data Assertion Variable
Join Point Strobe

Philipps-Universität Marburg

Faculty of Mathematics and Computer Science

Theses and Seminar Reports in the Programming Languages and Tools Group A Template and Guide

Christoph Bockisch, Stefan Schulz, and Teresa Dreyer

Seminar Report

March 06, 2025

Supervisor:

Prof. Dr.-Ing. C. Bockisch
*Programming Languages
and Tools Group*

Advisor:

M.Sc. Teresa Dreyer
*Programming Languages
and Tools Group*



Institution

Philipps-Universität Marburg
Faculty of Mathematics and Computer Science
Programming Languages and Tools Group
Hans-Meerwein-Str. 6
35043 Marburg
Deutschland

License

This work is licensed under a Creative Commons "" license.



Classification (ACM CCS 2012)

- **Applied computing** → **Education; Document preparation;**
- **Software and its engineering** → *Software notations and tools;*

Keywords

thesis, seminar paper, format template, structure template, good design

Abstract

The summary should provide a cross-section of the entire work, i.e. from the motivation to the conceptual approach, the execution and evaluation through to the result of the work. Only the most important points and conclusions should be briefly mentioned. For theses, an additional German Zusammenfassung is needed if the thesis is in English.

Contents

1	General information on Seminar Papers	1
2	Structure of the Report: Seminar Empirical Software Engineering	1
3	Structure of the Report: Seminar Debugging, Testing and Improving Programs	2
4	Literature.	2
5	Practical Tips for Using L^AT_EX	3
5.1	Hyphenation.	3
5.2	Quotation marks and dashes	3
5.3	TODOs.	4

5.4	Source Code Representation	5
5.5	Structure: Subsection.	6
5.6	Subsection.	7
5.7	References.	7
5.8	Bibliography.	8
5.9	Figures, Tables, Listings	8
5.10	LaTeX Packages Considerations.	9
6	References	10
A	A Famous Filler Text	11
B	Glossar	11

1 General information on Seminar Papers

The length of the written paper should be approximately 3000 words per person. This corresponds to about 6 pages. This does not include the cover sheet, bibliography and appendix. You can choose whether you want to write the paper in German or English.

If you have programmed as part of your seminar paper, please provide us with the code via git. Use the university gitlab for this. The same applies for your data and the complete script with your statistical analysis, if applicable. If you are dealing with sensitive data, please contact us. Do not upload them anywhere unless we have explicitly okayed this. You do not have to put the entire code in the appendix of the written paper, but can refer to the repository. The same applies for statistical analyses. Individual code examples in the body text should still be used where they make sense.

2 Structure of the Report: Seminar Empirical Software Engineering

The structure of the written paper generally depend on the actual seminar. This section is specific for the seminar “Empirical Software Engineering”. For hints on the structure for other seminars in the research group, in particular “Debugging, Testing and Improving Programs”, refer to the following section.

Your written paper should contain a short introduction and a short conclusion, if appropriate with an outlook. In the main part of your paper, you should explain the statistical analysis, briefly present the selected paper and critically examine it with regard to the statistical analysis. Also explain the conditions under which the method can/may be used and analyze whether the conditions for your paper are met. Please also refer to the slides from the kick-off meeting and any special features discussed individually that are relevant to your topic.

The presentation of the procedure should take up about half or less space. The critical analysis of the paper should be in the foreground, the summary of the paper should be kept as short as possible.

We do not prescribe the exact structure of the paper. For example, the structure could look like this

Reports and Theses in the PLT group

- Introduction
- Description of the Statistical Method
- Brief Summary of the Selected Paper
- Critical Analysis of the Selected Paper
- Conclusion

Attention: You need to adapt the template to reflect your topic and advisor.

3 Structure of the Report: Seminar Debugging, Testing and Improving Programs

Your written paper should contain a short introduction and a short conclusion, if appropriate with an outlook.

In the main part of your paper, you should summarize the seed paper that was provided to you and summarize your chosen additional paper. As part of this seminar, you are supposed to work out an example centered around a software error scenario to demonstrate the approach of one of the papers or both papers. This should also be discussed in the main part of the paper and should make up a substantial part of your report. Minimally, this should be the example used by the paper authors. Better would be your own example, if possible use the same example to demonstrate the approaches of both papers. Ideally you should practically use the tool or approach presented in the paper(s) and reproduce the evaluation. Your choices here will influence your grade, please also see the slides from the kick-off meeting. When in doubt, please contact your advisor.

Attention: You need to adapt the template to reflect your topic and advisor.

4 Literature

The majority of the sources used should be scientific, peer-reviewed sources. This means publications in conference proceedings or journals. Workshop publications typically have a lower status, but are still peer-reviewed.

Books are also often trustworthy sources and fully citable. However, there is not always a reviewing body for books, such as the peer review process for conferences, journals and workshops. Other sources, such as white papers, weblogs, documentation, etc. can also be used and referenced. For these sources in particular—but also for all sources you use in general—you must check for yourself whether you consider them trustworthy.¹ For example, you can distinguish between genuine publications (scientific articles and books) and other sources by including only the genuine publications in your bibliography. You can list the other sources as footnotes (see above). This also makes sense because these “other sources” are often only used in one place in the document. For specific information on reference management with LaTeX, see section Section 5.8.

¹see <http://www.findingdulcinea.com/news/education/2010/march/The-Top-10-Reasons-Students-Cannot-Cite-or-Rely-on-Wikipedia.html>.

5 Practical Tips for Using L^AT_EX

This section contains some practical tips for using L^AT_EX to structure your document and influence the appearance. It also presents some useful packages that can be used and points out common pitfalls.

5.1 Hyphenation

LaTeX automatically performs hyphenation using the rules for the language that you have specified as an option for the document class (see comments in the sample text file). If a word is not hyphenated correctly, you can define your own hyphenation for this word. A separate rule must always be defined for this. You can define new rules using the command `hyphenation{}` that apply throughout the rest of the document.

```
1 \hyphenation{su-per-cal-i-frag-i-lis-tic-ex-pi-al-i-do-cious}
2
3 I think this is: supercalifragilisticexpialidocious
```

I think this is: supercalifragilis-
ticexpialidocious

LaTeX can separate compound words with a hyphen automatically, except at the position of the hyphen:

Einnahmen-Überschuss-
Rechnung

Instead of the hyphen, use the command `\hyp{}` from the package `hyphenat` to also apply hyphenation to the partial words:

```
1 Einnahmen\hyp{Überschuss\hyp{Rechnung
```

Einnahmen-Überschuss-Rech-
nung

5.2 Quotation marks and dashes

If you want to use quotation marks, you must not simply use the corresponding ASCII symbol in LaTeX, as this has a control function. This document class offers the command `\quot{}` to set text in quotation marks. The quotation marks depend on the language setting that was passed to the class. The specified text itself can also use LaTeX commands. For example: `\quot{Auch der \textbf{Tüchtige} braucht \textit{Glück}.}` results in: “Auch der **Tüchtige** braucht *Glück*.”

LaTeX recognizes different forms of dashes: - (-) “hyphen or in German Bindestrich-Minus”, -- (–) “en-dash or in German Halbgeviertstrich”, --- (—) “em-dash, in German Geviertstrich”, each with different tasks. You can find a discussion of how to use these forms of strokes here:

- in English texts: <https://www.thepunctuationguide.com/index.html>
- in German texts:
 - <https://de.wikipedia.org/wiki/Geviertstrich>

Reports and Theses in the PLT group

- <https://de.wikipedia.org/wiki/Halbgeviertstrich>
- <https://de.wikipedia.org/wiki/Viertelgeviertstrich#Bindestrich-Minus>

5.3 TODOs

To insert TODOs or comments into the document, you can use the `todonotes` package, which is already loaded. See also <http://tug.ctan.org/macros/latex/contrib/todonotes/todonotes.pdf>. A list of all TODOs is automatically displayed at the beginning of the document so that you have an overview of whether you have processed all of them.

You can switch off this display of TODOs in the document by passing the option `notodo` to the document class in your LaTeX file: `\documentclass[ngerman,bachelor,notodo]{pltdoc}`. Then the overview at the beginning of the document is no longer displayed. You should use this option when you create the final version of your thesis.

You can insert a TODO comment using the command `\todo{Find a good example}`. Parameters can optionally be defined for the command in square brackets. For example, long comments are best inserted as inline comments, for which the inline parameter is specified: `\todo[inline]{This comment is much too long for the correction margin and would not be displayed legibly}`

Find a good example

This comment is much too long for the correction margin and would not be displayed legibly

You can use other parameters to define the color or the author of the comment. You can also define a short form of the comment that is displayed in the overview. Consult the package documentation for this.

To indicate that an image is missing at a certain point, you can use the command `\missingfigure{Here comes another illustrative image}`. The size of the placeholder can be determined via parameters.



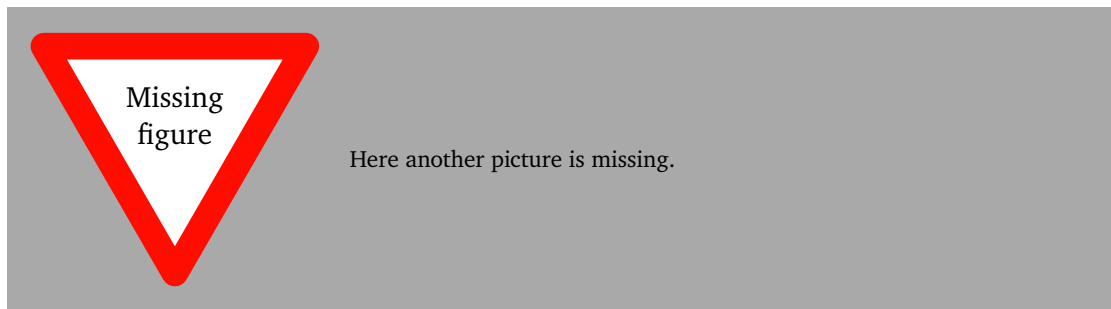
You can also use the command in a float environment. It is then treated like a normal figure (see Figure 1) and can later be replaced by `\includegraphics{}`

```
1 \begin{figure}
2 \missingfigure{Here, a picture is missing.}
3 \caption{Missing Picture}
4 \label{fig:missing}
5 \end{figure}
```

The LaTeX template defines a command for the creation of shortcuts for TODO macros. Regarding this, have a look at the LaTeX comment over the `\begin{document}` command.

Examples:

```
\cb{Test1}
\cb[Test2]{Test3}
\cbinline[Test2]{Test3}
```



■ **Figure 1** Missing Picture

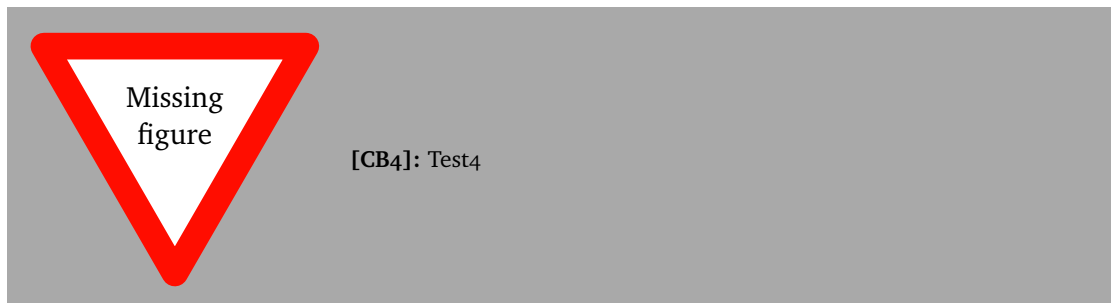
```
\cbfigure{Test4}
```

Test2 Test2

[CB3]: Test3

[CB1]:
Test1

[CB2]:
Test3



5.4 Source Code Representation

The class is based on the template provided by the Programming Journal (see <https://github.com/programming-journal/programming/releases>). The Programming Journal Template is based on the KOMA-Script Article (scrartcl, siehe <https://komascript.de/komascriptbestandteile>). This means that `\chapter` **not** is available. The commands `\frontmatter`, `\mainmatter` and `\backmatter` cannot be used either. The following structure levels are therefore available: `\section`, `\subsection`, `\subsubsection`, `\paragraph` and `\subparagraph`.

Use the listings package to display source code (<https://ftp.agdsn.de/pub/mirrors/latex/dante/macros/latex/contrib/listings/listings.pdf>). Minted is currently not correctly supported by this document class, for example the listings directory is not created correctly with Minted.

You can display source code via the environment `\begin{lstlisting} ... \end{lstlisting}` as a block (see Listing 1). You can either specify the language used in the source code as an optional parameter for each listing (`\begin{lstlisting}[language=Java] ...`) or use `\lstset{language=Java}` to set the default language from this command onwards (the language can be explicitly overwritten by specifying the optional parameter). In addition to displaying as a code block, you can also display formatted source code in continuous text. To do this, use the command `\lstinlineif(dndResult == null) {}!` (results in: `if(dndResult == null) {}`). The default language is also taken into account here and can also be specified explicitly using an optional parameter: `\lstinline[language=Java]!if(dndResult == null) {}!`. If the source text that you want to display contains an exclamation mark (!), you can also choose a different delimiter, e.g. `\lstinline|!` results in `!`.

Inline-Listings are not wrapped in an environment with justified text. You should therefore avoid long inline listings.

■ Listing 1 Example for Java source code with the listings package.

```

1 // Register a callback for the Mouse Up event
2 canvas.addListener(SWT.MouseUp, e -> {
3     synchronized (GUI.this) {
4         if (dndHandler != null && cardBeingMoved != null) {
5             Coordinate dndResult = dndHandler.cardDraggedAndDropped(cardBeingMoved,
6                 new Coordinate(e.x, e.y), new Coordinate(e.x - deltaX, e.y - deltaY));
7             if (dndResult != null) {
8                 cardBeingMoved.x = dndResult.x;
9                 cardBeingMoved.y = dndResult.y;
10            }
11        }
12        cardBeingMoved = null;
13        zoomedCard = null;
14        mouseUpCounter += 1;
15        canvas.redraw();
16        Logger.global.log(Level.INFO, "Mouse Up event");
17    }
18 });

```

You can use the German Umlauts ä, ö, ü, Ä, Ö, Ü und ß in Listings. If you want to use other special characters that the listings package cannot normally handle, write the following lines at the beginning of your document and add to the list of special characters. Add the following to “literate” for each special character: In curly brackets the character that you want to use in the listings environment, in curly brackets the command to generate the character, the character “r”. The output command must be enclosed in curly brackets again. The ‘r’ means that the inserted latex command stands for a character in the listing. For better clarity, you can separate the entries with spaces. You can also replace longer character strings, e.g. if you want to use formula characters or Greek letters in pseudocode. See also section 6.4 in <http://texdoc.net/texmf-dist/doc/latex/listings/listings.pdf>.

```

1 \lstset{
2     extendedchars=true,
3     literate={ä}{\a}1 {ö}{\o}1 {ü}{\u}1 {Ä}{\A}1 {Ö}{\O}1 {Ü}{\U}1 {ß}{\ss}1 {\@alpha}
4     { {\$alpha}{\$}}1,
5 }
6 if α then true else false

```

5.5 Structure: Subsection

This is the beginning of a section. The commands for structuring the text are:

1. \part{heading} (numbered) or \part*{heading} (not numbered)
2. \section{heading} (numbered) or \section*{heading} (not numbered)
3. \subsection{heading} (numbered) or \subsection*{heading} (not numbered)
4. \subsubsection{heading} (numbered) or \subsubsection*{heading} (not numbered)
5. \paragraph{heading} (numbered) or \paragraph*{heading} (not numbered)
6. \subparagraph{heading} (numbered) or \subparagraph*{heading} (not numbered)

If possible, do not use structures that are too deep. Typically, the structuring levels section – subsection are sufficient. You can also divide the work into parts using the part

structuring level. Only use this if your work deals with several independent focal points and the parts each have a certain size that justifies further subdivision.

Also pay attention to the balance of the structure. Very short sections or subsections should be avoided. Also, it is typically not good to have a single subsection in a section (this also applies to the other levels of structure). Below you will find examples of the structuring levels.

5.6 Subsection

This is a subsection.

5.6.1 Subsubsection

This is a subsubsection.

Paragraph

This is a paragraph.

Subparagraph

This is a sub paragraph.

5.7 References

To refer to other text passages in the text, you must first mark the target passage with a label. To do this, use the command `\label{thelabel}`. A label can be defined directly after a structure command (e.g. `\section{...}`). To mark graphics, tables, etc., define the label **behind** the `\caption{...}` command in the corresponding environment:

■ **Listing 2** Listing with an example of a label.

```
1 \begin{figure}
2 \begin{center}
3 \includegraphics[width=1.0\textwidth]{architecture.png}
4 \end{center}
5 \caption{Architektur-Diagramm}
6 \label{fig:architecture}
7 \end{figure}
```

You can define a label for a listing using an optional parameter. You must then also define a caption for the listing.

```
1 \begin{lstlisting}[language=TeX,
2   label={lst:example-label},
3   caption={Listing with an example of a label.}]
4 \begin{figure}
5 % ...
6 \end{figure}
7 \end{lstlisting}
```

To refer to the text passage with the label, you can use the command `\ref{lst:example-label}`. This generates the number of the element with the label: Listing 2. The command

Reports and Theses in the PLT group

`\autoref{lst:example-label}` creates a reference consisting of the type of object being referenced and its number: Listing 2. The name of the label is irrelevant. The distinction between the object types in the label (e.g. “lst:” for listings) is merely a convention.

5.8 Bibliography

To manage the literature used, use Bibtex (see <http://www.bibtex.org/Using/de/>). To do this, create an entry in a Bibtex file, e.g. “example.bin”, for each source you use. In the document, you can use `\cite{key}` to reference the source with the key “key”, for example `\cite{Georges2007}` [3]. You can use the optional parameter (in square brackets) to specify additional information, such as the chapter within a book, for a citation: `\cite[Section 4.10]{Gosling:2014}` results in [4:Section 4.10].

The bibliography is inserted into the document using the following commands:

```
1 \cleardoublepage
2 \bibliography{example}
3 \cleardoublepage
```

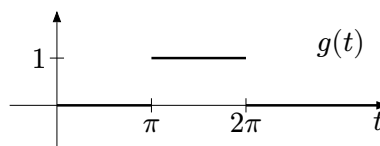
You can often download the entries for the Bibtex file or have them generated. For example, publishers such as ACM, Springer or IEEE offer the option of downloading the Bibtex entry for one of their articles. You can also use tools such as Mendeley (see <https://www.mendeley.com>) to manage your Bibtex file.

For references, use the `\textbackslash cite` command, e.g. [5], [6, 7] or [4:section 4.10].

5.9 Figures, Tables, Listings

Sometimes it is necessary to pay special attention to the *floating elements* of your document, that is figures, tables, listings, and similar. While it is tempting to force those elements to the specific position where they are written down, this is not always the best choice, especially since it can interrupt the reading flow. Therefore please refrain from using the `[h!]` and `[H]` specifiers for such elements. If you *must* have the element at the current position, it maybe is not a *floating element*, in the first place. It is perfectly fine to use graphics *without* a `\begin{figure}...\end{figure}` environment or use a `\begin{tabular}...\end{tabular}` *without* an enclosing `\begin{table}...\end{table}`.

The caption position of a floating element should match its content. For tables, listings, algorithms, or similar list-like content, please use captions *above* its contents, as those elements are typically read from top to bottom. Find examples in the listings already presented and Table 1. For figures, graphs, graphics, or similar, please use captions *below* the imagery, as in Figure 2.



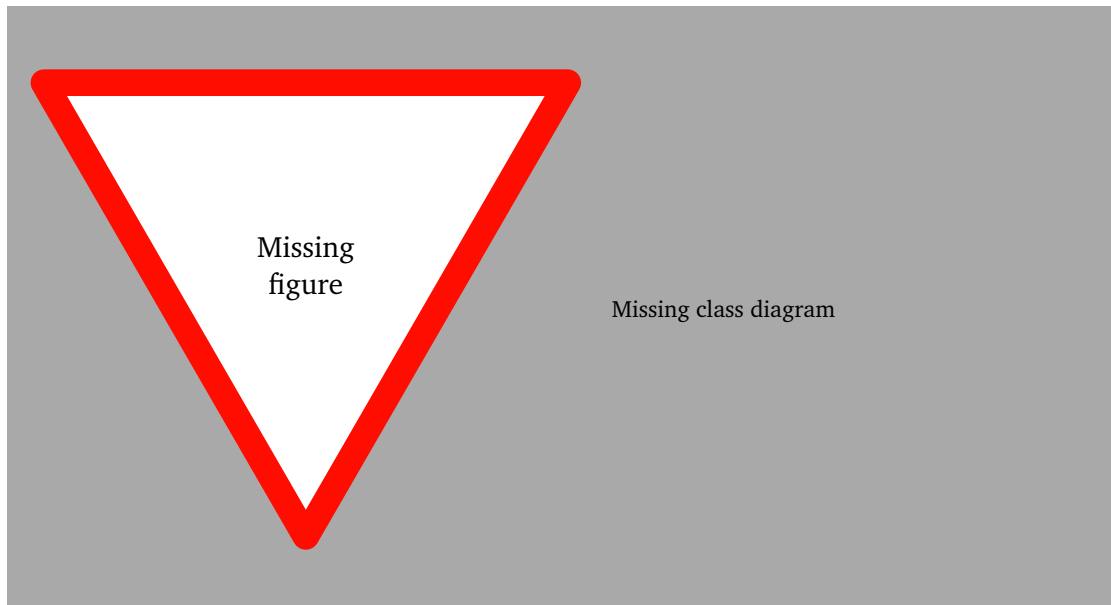
■ **Figure 2** A test figure

■ **Table 1** Differences between things projected and things achieved

Part	done	Value	Unit
Title	yes	2.3456	dB
Abstract	yes	10^3	khz
Rest is not entirely true			
Context	yes	90.473	%
Problem	no ^a		
Solution	yes	5642.5	MB
Implementation	yes	$1.2e-3$	m ² /s
Evaluation	no		
Related Work	no		
Conclusion	yes	4955.3	/kg

^a Just a few things missing

For tabular content, we suggest to avoid vertical lines altogether and keep horizontal lines to a minimum. Refer to the `booktabs` package documentation for more information [2]. Also, the `threeparttable` package is valuable if you need footnotes in your table [1].



5.10 LaTeX Packages Considerations

The following packages are automatically loaded by the `linline|programming|` class and need not to be loaded manually.

- `accsupp`
- `amsmath`
- `amstext`
- `array`
- `atbegshi`
- `babel`
- `booktabs`
- `calc`
- `caption`
- `colortbl`
- `comment`
- `csquotes`
- `doclicense`
- `expl3`
- `environ`
- `etoolbox`
- `FiraSans`
- `fnpct`
- `fontspec`
- `fontenc`

Reports and Theses in the PLT group

- | | | | |
|--------------------------------|-------------------------------|--------------------------|---------------------------|
| ▪ <code>getttitlestring</code> | ▪ <code>graphics</code> | ▪ <code>graphicx</code> | ▪ <code>grfext</code> |
| ▪ <code>ifthen</code> | ▪ <code>iflang</code> | ▪ <code>hypcap</code> | ▪ <code>hyphenat</code> |
| ▪ <code>hyperref</code> | ▪ <code>hyperxmp</code> | ▪ <code>listings</code> | ▪ <code>mathdesign</code> |
| ▪ <code>microtype</code> | ▪ <code>morewrites</code> | ▪ <code>multirow</code> | ▪ <code>multicol</code> |
| ▪ <code>relsize</code> | ▪ <code>rotating</code> | ▪ <code>siunitx</code> | ▪ <code>subcaption</code> |
| ▪ <code>tabularx</code> | ▪ <code>textcase</code> | ▪ <code>textcomp</code> | ▪ <code>tikz</code> |
| ▪ <code>tikz-uml</code> | ▪ <code>threeparttable</code> | ▪ <code>tocbibind</code> | ▪ <code>tocloft</code> |
| ▪ <code>typearea</code> | ▪ <code>url</code> | ▪ <code>verbatim</code> | ▪ <code>xcolor</code> |
| ▪ <code>xparse</code> | ▪ <code>xspace</code> | ▪ <code>xunicode</code> | |

The following packages **must not** be loaded.

- | | | | |
|----------------------------|------------------------------|--------------------------|--------------------------|
| ▪ <code>SIstyle</code> | ▪ <code>SIunits</code> | ▪ <code>a4wide</code> | ▪ <code>a4</code> |
| ▪ <code>aecompl</code> | ▪ <code>ae</code> | ▪ <code>caption2</code> | ▪ <code>courier</code> |
| ▪ <code>doublespace</code> | ▪ <code>epsfig</code> | ▪ <code>epsf</code> | ▪ <code>euler</code> |
| ▪ <code>fancyhdr</code> | ▪ <code>fancyheadings</code> | ▪ <code>fourier</code> | ▪ <code>geometry</code> |
| ▪ <code>glossary</code> | ▪ <code>helvet</code> | ▪ <code>isolatin</code> | ▪ <code>mathpple</code> |
| ▪ <code>mathptmx</code> | ▪ <code>mathptm</code> | ▪ <code>newtxmath</code> | ▪ <code>newtxtext</code> |
| ▪ <code>palatino</code> | ▪ <code>psfig</code> | ▪ <code>pslatex</code> | ▪ <code>scrpage</code> |
| ▪ <code>subfigure</code> | ▪ <code>subfig</code> | ▪ <code>trenc</code> | ▪ <code>times</code> |
| ▪ <code>umlaut</code> | ▪ <code>utopia</code> | ▪ <code>zefonts</code> | |

Additionally, any package that changes the font must not be loaded.

6 References

- [1] Donald Arseneau. 2003. Tables with captions and notes all the same width.
- [2] Danie Els and Simon Fear. 2016. Publication quality tables in LaTeX.
- [3] Andy Georges, Dries Buytaert, and Lieven Eeckhout. 2007. Statistically rigorous java performance evaluation. In *Proceedings of the 22nd Annual ACM SIGPLAN Conference on Object-Oriented Programming, Systems, Languages, and Applications, OOPSLA 2007, October 21-25, 2007, Montreal, Quebec, Canada, 2007*. ACM, 57–76. <https://doi.org/10.1145/1297027.1297033>
- [4] James Gosling, Bill Joy, Guy L. Steele, Gilad Bracha, and Alex Buckley. 2014. *The Java Language Specification, Java SE 8 Edition* (1st ed.). Addison-Wesley Professional.
- [5] Sharath Gude, Munawar Hafiz, and Allen Wirfs-Brock. 2014. Javascript: The used parts. In *Computer Software and Applications Conference (COMPSAC), 2014 IEEE 38th Annual*, 2014. 466–475.
- [6] Martin Odersky and al. 2004. *An Overview of the Scala Programming Language*.
- [7] Hartmut Schlosser. Alle Java 9 Features erklärt: Oracle veröffentlicht JDK9-Dokumentation.

Appendix

A A Famous Filler Text

Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut enim ad minim veniam, quis nostrud exercitation ullamco laboris nisi ut aliquip ex ea commodo consequat. Duis aute irure dolor in reprehenderit in voluptate velit esse cillum dolore eu fugiat nulla pariatur. Excepteur sint occaecat cupidatat non proident, sunt in culpa qui officia deserunt mollit anim id est laborum.

B Glossar

If it makes sense in your case, you can include a glossary. Please be aware that the \LaTeX package glossary is not compatible with the document class and can not be used or included.