The mluexercise class*

Jan Heinrich Reimer
Fachschaftsrat Mathematik/Informatik †
https://fachschaft.mathinf.uni-halle.de
fachschaft@mathinf.uni-halle.de
Released 2020/11/12

Contents

1 Introduction 1

2 Usage . 2
2.1 Exercise Metadata . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 2
2.2 Student Metadata . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 2
2.3 Included Packages . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 3

3 Design Goals 3

4 Implementation 3
4.1 Setup . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 3
4.2 Global Variables . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 5
4.3 Layout, Text . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 5
4.4 Mathematics . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 7
4.5 Algorithms, Pseudocode . . . . . . . . . . . . . . . . . . . . . . . . . . . 9
4.6 Code Listings . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 10
4.7 Graphics, Drawing, Plots . . . . . . . . . . . . . . . . . . . . . . . . . . . 11
4.8 Document Hooks . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 11

1 Introduction

The mluexercise class is a template class for solving weekly exercises at the Institute for Computer Science of Martin Luther University Halle-Wittenberg. The class can be used by all students—especially first semesters—to typeset their exercises with a low-effort in beautiful \LaTeX. We include a bunch of handy macros that are used throughout many lectures during the bachelor’s degree program.

*This document describes version v2.0, last revised 2020/11/12.
†Fachschaftsrat Mathematik/Informatik der Studierendenschaft der Martin-Luther-Universität HalleWittenberg K.d.o.R., Von-Seckendorff-Platz 1, 06120 Halle (Saale), Germany
‡https://uni-halle.de/
2 Usage

\documentclass Load the mluexercise class at the start of your preamble.
\documentclass{mluexercise}

Languages As in many classes, you can define your main document language. This will help \LaTeX in hyphenating words and structural texts (e.g., sections) are being translated automatically.

english The english option selects English language for texts and hyphenation.
ngerman The ngerman option selects German language (with new spelling) for texts and hyphenation.

While you do not have to specify a language, it is a meaningful information for writers (including yourself) to explicitly state the desired language that should be used consistently throughout the whole document. If no language is specified, English language is used.

dataminingstyle Styles The dataminingstyle option may be used for the "Data Mining" lecture and overwrites some mathematical symbols to match the lecture’s definition’s.

2.1 Exercise Metadata

Define the exercise’s metadata and your identity. The below macros should be used only in the preamble, that is before \begin{document}.
\begin{itemize}
\item lecture The lecture{⟨name⟩} macro sets the name of the lecture you are writing exercises for.
\item semester The semester is automatically set to the current semester. You can overwrite the default value with the semester{⟨semester⟩} macro if you are compiling exercises for another semester.
\item exercise The exercise{⟨number⟩} macro sets the number of the exercise you are working on.
\item task The task{⟨number⟩} macro sets the number of the first task. This is useful if you submit single PDF files for each task within the exercise. If you omit this option, numbering starts at 1.
\item group Sometimes tutors allow to submit exercises in groups. You can use the optional group{⟨number⟩} macro to set the group number. If no group number is given, the group number is not displayed.
\item title The title{⟨lecture name⟩} macro is an alias for \lecture{⟨lecture name⟩}.
\item date The date is automatically set to the current date. You can overwrite the default value with the date{⟨date⟩} macro.
\end{itemize}

2.2 Student Metadata

The following commands should be used to include identification in your exercise in order to assign your work.
\begin{itemize}
\item studentname You must define your full name with the studentname{⟨full name⟩} macro. If you are working in a group, you can enter more names separated with commas and/or line breaks.
\item studentnumber The studentnumber{⟨matriculation number⟩} macro specifies your student ma-
triculation number (e.g., 234 567 890). This is the same number as on your student ID card.

\studentsymbol\{\langle shorthand symbol\rangle\} macro specifies your shorthand symbol used for university IT services (e.g., abcde). This is the same username used for Stud.IP, university mail etc.

Both \studentnumber and \studentsymbol are optional. Though, most tutors require specifying either student number or symbol.

Multiple Students While you can use commas and line breaks in the above macros, we would suggest one of the following two options to submit work in groups with multiple students:

- Either use the services offered by the university to check-in your group members online (e.g., in the Institute for Computer Science’s "Übungsportal"). In this case you don’t need to specify multiple names in the preamble, but just your own name.
- Or, if you cannot specify group members elsewhere, we recommend to add a footnote on the first page. That way you can list all other students but do not waste too much paper.

2.3 Included Packages

The \mluexercise class loads a number of useful \LaTeX\ packages listed in Table 1. Those are needed to typeset exercises during the Computer Science bachelor’s degree programme at Martin Luther University Halle-Wittenberg. We recommend reading the introductions of each package’s documentation on CTAN.

3 Design Goals

The \mluexercise class should:

- be usable for first semester students with rather little \LaTeX\ experience,
- be consistent with Martin Luther University brand guidelines especially with respect to font families and colors, and
- include packages frequently needed by students.

4 Implementation

4.1 Setup

We specify the \LaTeX\ version, define and parse options and then load the base class \texttt{scrartcl} with all remaining options.

\NeedsTeXFormat{LaTeX2e}
\ProvidesPackage{mluexercise}[2020/11/12 v2.0]
\url{https://ctan.org/}
\url{https://www.pr.uni-halle.de/download/logo/}
\url{https://ctan.org/pkg/scrartcl}
Table 1: Packages included in the class.

<table>
<thead>
<tr>
<th>Package</th>
<th>Purpose</th>
<th>Link</th>
</tr>
</thead>
<tbody>
<tr>
<td>babel</td>
<td>hyphenation, localization</td>
<td><a href="https://ctan.org/pkg/babel">https://ctan.org/pkg/babel</a></td>
</tr>
<tr>
<td>amsmath</td>
<td>mathematics</td>
<td><a href="https://ctan.org/pkg/amsmath">https://ctan.org/pkg/amsmath</a></td>
</tr>
<tr>
<td>amsthm</td>
<td></td>
<td><a href="https://ctan.org/pkg/amsthm">https://ctan.org/pkg/amsthm</a></td>
</tr>
<tr>
<td>amssymb</td>
<td></td>
<td><a href="https://ctan.org/pkg/amssymb">https://ctan.org/pkg/amssymb</a></td>
</tr>
<tr>
<td>amstext</td>
<td></td>
<td><a href="https://ctan.org/pkg/amstext">https://ctan.org/pkg/amstext</a></td>
</tr>
<tr>
<td>array</td>
<td></td>
<td><a href="https://ctan.org/pkg/array">https://ctan.org/pkg/array</a></td>
</tr>
<tr>
<td>cancel</td>
<td></td>
<td><a href="https://ctan.org/pkg/cancel">https://ctan.org/pkg/cancel</a></td>
</tr>
<tr>
<td>inputenc</td>
<td>UTF8 encoding</td>
<td><a href="https://ctan.org/pkg/inputenc">https://ctan.org/pkg/inputenc</a></td>
</tr>
<tr>
<td>fontenc</td>
<td>T1 encoding</td>
<td><a href="https://ctan.org/pkg/fontenc">https://ctan.org/pkg/fontenc</a></td>
</tr>
<tr>
<td>booktabs</td>
<td>tables</td>
<td><a href="https://ctan.org/pkg/booktabs">https://ctan.org/pkg/booktabs</a></td>
</tr>
<tr>
<td>graphicx</td>
<td>images, graphics</td>
<td><a href="https://ctan.org/pkg/graphicx">https://ctan.org/pkg/graphicx</a></td>
</tr>
<tr>
<td>xcolor</td>
<td>color definitions</td>
<td><a href="https://ctan.org/pkg/xcolor">https://ctan.org/pkg/xcolor</a></td>
</tr>
<tr>
<td>tikz</td>
<td>drawing vector graphics</td>
<td><a href="https://ctan.org/pkg/tikz">https://ctan.org/pkg/tikz</a></td>
</tr>
<tr>
<td>pgfplots</td>
<td>plotting</td>
<td><a href="https://ctan.org/pkg/pgfplots">https://ctan.org/pkg/pgfplots</a></td>
</tr>
<tr>
<td>algorithm2e</td>
<td>pseudocode, algorithms</td>
<td><a href="https://ctan.org/pkg/algorithme2e">https://ctan.org/pkg/algorithme2e</a></td>
</tr>
<tr>
<td>listings</td>
<td>code listings</td>
<td><a href="https://ctan.org/pkg/listings">https://ctan.org/pkg/listings</a></td>
</tr>
<tr>
<td>listingsutf8</td>
<td>sub-figures</td>
<td><a href="https://ctan.org/pkg/listingsutf8">https://ctan.org/pkg/listingsutf8</a></td>
</tr>
<tr>
<td>subcaption</td>
<td>quotation</td>
<td><a href="https://ctan.org/pkg/subcaption">https://ctan.org/pkg/subcaption</a></td>
</tr>
<tr>
<td>csquotes</td>
<td>quotation</td>
<td><a href="https://ctan.org/pkg/csquotes">https://ctan.org/pkg/csquotes</a></td>
</tr>
</tbody>
</table>

Options

\newif\ifdataminingstyle
\DeclareOption{dataminingstyle}{% \dataminingstyletrue}
\ProcessOptions*
\LoadClass{scrartcl}

To enable support for UTF8 and some other useful basics, we load a bunch of packages.

\RequirePackage[utf8]{inputenc}
\RequirePackage[T1]{fontenc}
\RequirePackage{ifthen} % Conditional branches and loops.
\RequirePackage{etoolbox} % Hooks for executing code.
\RequirePackage{hyperref} % Hyperlinks.
\RequirePackage{xcolor} % Color definitions.

As we’d like to be able to switch between English and German with proper hyphenation, load language support packages.\footnote{If you’d like to add a language, please contact us.}

\RequirePackage{babel}
\RequirePackage{iflang}
\newcommand{\IfGerman}[2]{% \IfLanguagePatterns{german}{#1}{% \IfLanguagePatterns{ngerman}{#1}{#2}}}

\IfGerman{IfGerman}{\IfGerman}{german}{\IfGerman}{ngerman}{#1}{#2}}
4.2 Global Variables

The class can be configured with the lecture's and student's details. We define global variables for that and redefine some built in \LaTeX commands to ease the migration from standard classes.

20 \newcommand{\@checkoption}[3]{%
21 \AtEndPreamble{
22 \ifthenelse{\equal{\the#1}{} }{
23 \ClassError{mluexercise}{Missing #2}{%
24 \protect\#2 using the \protect\#3 command.}%
25 %}
26 }{}
27 \}%
28 }
29 \newtoks{\@lecture}
30 \newcommand{\lecture}[1]{\global{\@lecture}{#1}}
31 \let{\title}{\lecture} % Redefine standard \title command.
32 \@checkoption{\@lecture}{lecture name}{\lecture}
33 \newtoks{\@semester}
34 \newcommand{\semester}[1]{\global{\@semester}{#1}}
35 \global{\@semester}{ % Automatically set semester based on current date.
36 \ifnum\month<4 % Winter semester, including previous year.
37 \WS{\the\year} \else \ifnum\month<10 % Summer semester.
38 \SS{\the\year} \else % Winter semester, including next year.
39 \WS{-1}{\the\year}\fi\fi
40 \}%
41 \newtoks{\@exercise}
42 \newcommand{\exercise}[1]{\global{\@exercise}{#1}}
43 \@checkoption{\@exercise}{exercise number}{\exercise}
44 \newtoks{\@task}
45 \newcommand{\task}[1]{\global{\@task}{#1}}
46 \global{\@task}{1}
47 \newtoks{\@group}
48 \newcommand{\group}[1]{\global{\@group}{#1}}
49 \global{\@group}{0}
50 \newtoks{\@studentname}
51 \newcommand{\studentname}[1]{\global{\@studentname}{#1}}
52 \let{\author}{\studentname} % Redefine standard \author command.
53 \@checkoption{\@studentname}{student name}{\studentname}
54 \newtoks{\@studentnumber}
55 \newcommand{\studentnumber}[1]{\global{\@studentnumber}{#1}}
56 \newtoks{\@studentsymbol}
57 \newcommand{\studentsymbol}[1]{\global{\@studentsymbol}{#1}}

4.3 Layout, Text

Set up a reasonable page layout and define text styles that comply with Martin Luther University brand guidelines.

61 \RequirePackage{calc}
62 \RequirePackage{a4paper, inner=2cm, outer=2cm, top=3cm, bottom=3cm, head=0.75cm, headsep=0.25cm}
Font Families We use Libertine as main font as that is the closest free font to the proprietary URW Classico font used by the university. For mathematics we use the more modern Euler Maths font and for source code the true monospace font Source Code Pro, as otherwise copying from the PDF is broken. For clarity reasons we do not want code ligatures.

Text Styles

Text Styles

URL / Link Font

Headings

Title
4.4 Mathematics

We rely on AMS packages for rendering mathematical equations, proofs, and symbols. Additionally we load packages for arrays in math mode and cancellations.

Now we define (more readable) macros for the mathematical symbols and functions most commonly used at Martin Luther University.

\newcommand{\union}{\cup}
\newcommand{\disjunction}{\uplus}
\newcommand{\intersection}{\cap}
\newcommand{\intersect}{\cap}
\newcommand{\infinity}{\infty}
\newcommand{\corresponds}{\triangleq}
\newcommand{\C}{\mathbb{C}} \% Complex numbers.
\newcommand{\realnumbers}{\R}
\newcommand{\Q}{\mathbb{Q}} \% Rational numbers.
\newcommand{\Z}{\mathbb{Z}} \% Whole numbers.
\newcommand{\N}{\mathbb{N}} \% Natural numbers.
\newcommand{\B}{\mathbb{B}} \% Binary numbers.
\newcommand{\eqtransform}{\ensuremath{\qquad\big|\,\,}}
\% Equivalence transformation
\newcommand{\ditto}{\ensuremath{\quad\quad}}
\% "Ditto" sign
Many maths symbols can have alternatives for stylistic choices. We enable the variants most liked by our tutors.

For some specific lectures we provide more detailed macros for symbols and functions.

Lecture “Lineare Algebra”
\begin{Verbatim}
\newcommand{\base}{\mathcal{#1}} % Base (calligraphic)
\DeclareMathOperator{\im}{im} % Image
\DeclareMathOperator{\id}{id} % Identity
\DeclareMathOperator{\sel}{sel} % Selection
\DeclareMathOperator{\dom}{dom} % Domain
\DeclareMathOperator{\ran}{ran} % Range
\DeclareMathOperator{\Hom}{Hom} % Homomorphism
\DeclareMathOperator{\End}{End} % Endomorphism
\end{Verbatim}

Lecture “Datenstrukturen und effiziente Algorithmen”
\begin{Verbatim}
\renewcommand{\O}{\mathcal{O}} % asymptotic O-Notation (Landau)
\DeclareMathOperator{\indeg}{indeg} % Indegree
\DeclareMathOperator{\outdeg}{outdeg} % Outdegree
\end{Verbatim}

Lecture "Einführung in Data Science"
\begin{Verbatim}
\renewcommand{\P}{\ifdataminingstyle p\else\mathbf{P}\fi}
\newcommand{\E}{\ifdataminingstyle\mathbb{E}\else\mathbf{E}\fi}
\DeclareMathOperator{\var}{var}
\DeclareMathOperator{\Var}{Var}
\DeclareMathOperator{\cov}{cov}
\DeclareMathOperator{\Cov}{Cov}
\DeclareMathOperator{\Bin}{Bin}
\DeclareMathOperator{\Exp}{Exp}
\DeclareMathOperator{\Dir}{Dir}
\DeclareMathOperator{\Mult}{Mult}
\newcommand{\Normal}{\mathcal{N}}
\newcommand{\Norm}{\Normal}
\end{Verbatim}

Table Columns  The handy \texttt{tabular} column definitions below can be used to typeset a whole column in math mode, to avoid repetetive opening/closing brackets.
\begin{Verbatim}
\newcolumntype{L}{>{\(}l<{\)}}
\newcolumntype{R}{>{\(}r<{\)}}
\newcolumntype{C}{>{\(}c<{\)}}
\end{Verbatim}

Calculus Environments  Calculus environments are currently work in progress, subject to change, and may not work as expected. However, they can simplify writing logical consequences in the “Mathematische Grundlagen der Informatik und Konzepte der Modellierung” lecture.
\newcounter{calculusRowCount}
\RequirePackage{pgfkeys}
\newenvironment{calculus}{\pgfkeys{/mlu/calculus/.cd,show index=false,context=\Gamma,context command=context,#1}}{
\setcounter{calculusRowCount}{0}
\newcommand{\calculusSymbol}{\text{sffamily\itshape}\pgfkeysvalueof{/mlu/calculus/symbol}}
\newcommand{\calculusContext}{\ensuremath{\pgfkeysvalueof{/mlu/calculus/context}}}
\expandafter\let\csname \pgfkeysvalueof{/mlu/calculus/context command}\endcsname\calculusContext
\par\vspace{0.5em}\begin{minipage}{\textwidth}\begin{tabular}{@{\stepcounter{calculusRowCount}\flushright}R}
\vdash_{\text{%}}%\ifthenelse{\equal{\pgfkeysvalueof{/mlu/calculus/show index}}{true}}{\ifthenelse{\equal{\pgfkeysvalueof{/mlu/calculus/symbol}}{}%}{\calculusSymbol}%}{%\calculusSymbol}%}{\calculusSymbol}%\hspace{-0.1em}%\calculusContext \calculusSymbol\hspace{-0.1em}}\end{tabular}\end{minipage}\vspace{0.5em}\let\calculusContext\undefined \let\calculusSymbol\undefined\end{calculus}}
\newenvironment{eqcalc}{\begin{calculus}\begin{array}{l}}{\end{calculus}} % Equation calculus
\newenvironment{seqcalc}{\begin{calculus}\begin{array}{l}}{\end{calculus}} % Sequence calculus

4.5 Algorithms, Pseudocode

\RequirePackage[vlined,linesnumbered]{algorithm2e}
\SetKwProg{Function}{function}{\ is}{end function}
\SetKwComment{Comment}{\quad\triangleright~}{} % Comment style.
\SetCommentSty{itshape} % Comment font.
\SetKw{Continue}{continue}
\SetKwBlock{Repeat}{repeat}{}} % Line number font.
\SetAlFnt{\footnotesize} % Line number skip.
\SetAlGmInRelativeSize{0}
\SetAllFnt{\footnotesize}
4.6 Code Listings

For syntax highlighting and including source code listings, we use the `listings` package. With `listingsutf8` we can include listings that contain Unicode characters.

```latex
\RequirePackage{listings}
\RequirePackage{listingsutf8} % UTF8 support in included listings.
\definecolor{lsnumber}{rgb}{0,0,0} % Zeilennummerfarbe
\definecolor{lscomment}{rgb}{0.25,0.5,0.35} % Kommentarfarbe
\definecolor{lskeyword}{rgb}{0.5,0,0.35} % Schlüsselwörterfarbe
\definecolor{lsstring}{rgb}{0.6,0,0} % Zeichenkettenfarbe
```

The default listing style below is enabled automatically and should guarantee consistency with the appearance of pseudocode (see Section 4.5).

```latex
\lstset{
  language=C,
  basicstyle=\ttfamily,
  breakatwhitespace=false,
  breaklines=true,
  prebreak=\\footnotesize\hookleftarrow,
  numbers=left,
  numberstyle=\color{lsnumber}\tiny,
  numbersep=0.5em,
  stepnumber=1,
  captionpos=b,
  emphstyle={\bfseries},
  showstringspaces=false,
  tabsize=2,
}
```

Haskell Language Definition

Refine the Haskell language definition and code style for use in the "Konzepte der Programmierung" lecture.

```latex
\lstdefinelanguage[Zimmermann]{haskell}{haskell}{
  escapeinside={*'}{'*},
  showstringspaces=false,
  morecomment=[s][\color{lscomment}]{//}{*/,
  keepspaces=true,
  keywordstyle=\bfseries\color{lskeyword},
  stringstyle=\color{lsstring},
  showtabs=false, showspaces=false,
  showstringspaces=false,
  tabsize=2,
}
```

Assembler Language Definition

Introduce a new language definition for Prof. Molitor’s Assembler dialect used in "Einführung in Rechnerarchitektur".

```latex
\lstdefinestyle[Assembler]{x86masm}{Assembler}{
  morekeywords={
    ldd,sto,shl,shr,rol,ror,sub,add,shr,rol,ror1, subi,addi,or,and,xor,xnor,jmp,beq,bneq,bgt,bo,ldpc,stpc
  },
  comment=[l]{\#},
}
```

Haskell Language Definition (continued)

```latex
\lstdefinestyle{haskell}{language=zhaskell}
```

Assembler Language Definition (continued)

```latex
\lstalias{}{zhaskell}{Zimmermann}{haskell}
% Deprecated. Use 'language=zhaskell' instead of 'style=haskell'.
\lstdefinestyle{haskell}{language=zhaskell}
```
4.7 Graphics, Drawing, Plots

We add packages for including images and graphics, for drawing vector graphics, and for plotting data or functions.

\RequirePackage{booktabs}
\RequirePackage{graphicx}
\RequirePackage{float}
\RequirePackage{subcaption}
\RequirePackage{tikz}
\RequirePackage{pgfplots}
\RequirePackage{rotating}
\usetikzlibrary{positioning}
\usetikzlibrary{automata}
\usetikzlibrary{trees}
\tikzset{
>=latex,
font=sffamily,
}
\pgfplotsset{compat=1.16}

4.8 Document Hooks

The following \LaTeX code should be inserted just before \begin{document}.

\AtEndPreamble{
Update exercise and task number as specified in the preamble. (Set first section number to task number.)
\setcounter{section}\numberline{}\addtocounter{section}{-1}
\newcommand{\@exercisestring}{\IfGerman{\the\@exercise.~Übungsserie}{Exercise~\the\@exercise}}
\ihead{\the\@studentname}
\chead{\textbf{\@exercisestring}}
\ohead{\the\@studentnumber%
\ifthenelse{\equal{\the\@studentnumber}{} \OR\equal{\the\@studentsymbol}{}\slash\slash}{\the\@studentsymbol}}
\set up page head/foot with metadata.
\hypersetup{
pdfauthor=\the\@studentname,
pdftitle={\@exercisestring - \the\@lecture}}

\set up PDF metadata.
\hypersetup{
}
}