

The HomeworkAssignment class*

Adrian C Hinrichs
adrian.hinrichs@rwth-aachen.de

May 19, 2017

1 Abstract

This class provides a relative simple document-type for homework, mainly created for assignments at the University. This class is inherited from `article`, it is not perfect, but I am trying my very best.

2 Options

<code>problemstyle=<1></code>	These options allow the customization of the displayed numbers. For Example, if
<code>subproblemstyle=<1></code>	<code>problemstyle=Roman</code> , <code>subproblemstyle=arabic</code> , <code>subsubproblemstyle=roman</code>
<code>subsubproblemstyle=<1></code>	is passed, The first subsubproblem of the first subproblem of the first problem would be labeled as i) of Problem I.1 .
	Available options are <code>arabic</code> , <code>Alph</code> , <code>alph</code> , <code>Roman</code> , and <code>roman</code> . Standard values are: <code>problemstyle=arabic</code> , <code>subproblemstyle=alph</code> , <code>subsubproblemstyle=roman</code> .
<code>design=<1></code>	Allows the User to select an older page-style, for backwards compatibility. Recognized values are <code>v1</code> and <code>v2</code> . Everytime a version changes the default look, a new possible value will be added.
	Only set this if you really need to get an old look, <i>older styles are not going to be maintained!</i>

2.1 Inherited options

Because the class is inherited by `article`, every Option that can be passed to `article`, will be passed to `article`.

3 Commands

3.1 Document Informations

<code>\subject</code>	Sets the subject of the document. Takes the subject as argument. Standard Value
<code>\kurs</code>	<hr/> *This document corresponds to HomeworkAssignment v2.0,dated 2017/05/13.

is “Kein Kurs”
`\kurs` is deprecated.

`\tutorial` Sets the tutorial of the author. Takes it as an argument. Standard Value is empty, so that this command can be omitted.
`\tutorium` is deprecated.

`\deadline` Sets the deadline of the document. Takes it as an argument. Standard value is `\today`.
`\abgabe` is deprecated

3.1.1 Inherited from article

`\author` Sets the author of the document.
`\date` Sets the date of the document.

3.2 Sectioning

Because the class is designed for Assignments, the sectioning-commands are different from Article

3.2.1 ‘plain’ Sectioning

`\problem` These commands work like their counterpart in article, except that there will be no number, nor will they increase a counter. Nevertheless, they will be shown in the table of contents.
`\subproblem`
`\subsubproblem`

`\solution` They work like Paragraph, but do not take an argument, instead they print out “Lösung”, “Beweis”, “Gegeben”, and “Zu zeigen”, respectively¹. They are not mentioned in the table of contents.
`\proof`
`\given`
`\toShow`

3.2.2 ‘better’ Sectioning

The following commands are an augmented version of the “plain” commands.

`\newproblem` These commands require no argument, and automatically create a numbered title. The optional Argument is the new value for the corresponding counter.
`\newsproblem`
`\newsproblem`
`\newsproblem`

3.3 Useful Macros

3.3.1 Quod Erat Demunstarndum, End of Proof

`\QED` Display a flushed-right *QED*, \square , or \blacksquare , respectively. `\qed` is not implemented, to keep compatibility to several Math-packages, which define the later.
`\EOP`

`\eop` ¹As of v1.6, Translations are added, depending on the chosen Language, there may be an other Text displayed.
See 8.2 for all Translations

3.3.2 Quod Non Erat Demunstarndum at iucundum est

`\QNED` Display a flushed-right \triangle . `\QNED` displays it in a new line, `\qned` at the end of the same line.

In Mathematical proofs this symbol is used to mark things, which we did not intend to proof, but are interesting anyway.

3.4 Stolen Goods

»Das ist alles nur geklaut«

~Tobias Künzel

These Commands are not mine, they are all stolen from Alexander Bartolomey's² `amath-Class`³

⁴⁵ Defines a set of mathematical fields, which are very usefull (see Table 1)

	Command	Output	Description
<code>\N</code>	<code>\N</code>	\mathbb{N}	Natural Numbers
<code>\Z</code>	<code>\Z</code>	\mathbb{Z}	Whole? Numbers
<code>\Q</code>	<code>\Q</code>	\mathbb{Q}	Rational Numbers
<code>\C</code>	<code>\C</code>	\mathbb{C}	Complex Numbers
<code>\F</code>	<code>\F</code>	\mathbb{F}	Prime-Field?
<code>\Primes</code>	<code>\Primes</code>	\mathbb{P}	Set of all Primes

Table 1: Field-Commands

`\GL` Output usefull Mathematical functions and Operators.

`\id` All, except `\dif{<1>}` print the 'themselves' as text, `\diff{<1>}` outputs $\frac{d}{d<1>}$.

`\diff{<1>}`

`\dx`

4 Pagestyle

4.1 Headers

²“Occloxiium” on GitHub:<https://github.com/occloxiium>

³`Amath.sty` is part of Alexander Bartolomey's Alphabet Classes: <https://github.com/occloxiium/AlphabetClasses>

⁴Has to be \mathbb{P} , because \mathbb{P} is already in use

⁵not a Field

5 Development and support

The package is developed at *github*:

<https://github.com/ACHinrichs/LaTeX-templates>

Please refer to that site for any bug report or development information.

6 Changelog

v1.0 - 2016/10/23 Intial

v1.1 - 2016/11/02 ...

v1.2 - 2016/11/03 ...

v1.3 - 2016/12/01 Provide the Class as .dtx file and more

v1.4 - 2017/04/29 “Minor” bugfixes

v1.5 - 2017/04/29 Problems are displayed in the table of contents. Type of numeration is now configurable.

v1.5.1 - 2017/04/29 Bugfix

v1.5.2 - 2017/04/29 Add version-number

v1.6 - 2017/05/02 Add Translations (German and English)
Add `\given` and `\toShow`
Add `\QED`, `\EOP`, and `\eop`

v1.6.3 - 2017/05/05 Bugfixes

v1.6.4 - 2017/05/09 Change `\eop` to be in the same line

v1.7 - 2017/05/09 Add `\QNEd`

v2.0 - 2017/05/09 Change Margins,
Add Option to select older Page-Style,
Change standardlayout to twocolumn and twoside

7 Examples

For examples, please see the Git-Repo at <https://github.com/ACHinrichs/LaTeX-templates>

8 Implementation

The following part is very boring, but I have not found a solution to create a .cls-file without including the implementation into the document. Loads L^AT_EX₂ε and sets the Version Loads the `article`, which is the base-class.

8.1 Packages & Options

```
1 \RequirePackage{kvoptions}
2 \SetupKeyvalOptions{ family=hwa,
3   prefix=hwa@ }
4 \DeclareStringOption[arabic]{problemsty}
5 \DeclareStringOption[alph]{subproblemsty}
6 \DeclareStringOption[roman]{subsubproblemsty}
7 \DeclareBoolOption[false]{listings}
8 \DeclareStringOption[v2]{design}
9 \DeclareBoolOption[true]{twoside}
10 \DeclareComplementaryOption{oneside}{twoside}
11 \DeclareBoolOption[true]{twocolumn}
12 \DeclareComplementaryOption{onecolumn}{twocolumn}
13 % Redefine the article-options
14 % \begin{macrocode}
15 \DeclareDefaultOption{\PassOptionsToClass{\CurrentOptionKey}{article}}

    Processes the Options and loads article
16 \ProcessKeyvalOptions*
17 \ifhwa@twoside
18 \PassOptionsToClass{twoside}{article}
19 \else
20 \PassOptionsToClass{oneside}{article}
21 \fi
22 \ifhwa@twocolumn
23 \PassOptionsToClass{twocolumn}{article}
24 \else
25 \PassOptionsToClass{onecolumn}{article}
26 \fi
27 \LoadClass{article}

    Loads required Packages
28 \RequirePackage{suffix}
29 \RequirePackage{fancyhdr}
30 \RequirePackage{ifthen}
31 \RequirePackage{translations}
32 \RequirePackage{amssymb}
```

```

33 \ifhwa@listings
34 \RequirePackage{listings}
35 \lstset{
36   frame = single,
37   breaklines = true,
38   postbreak=\raisebox{0ex}[0ex][0ex]{\ensuremath{\hookrightarrow}\space}},
39   basicstyle=\scriptsize
40 }
41 \else
42 \empty
43 \fi
44
45 % Make sure that this is the last Package loaded
46 \ifthenelse{\equal{\hwa@design}{v2}}{
47   \RequirePackage{geometry}
48   \ifhwa@twocolumn
49     \geometry{top=2cm, bottom=2cm, left=2cm,
50       headsep=14pt,hmarginratio={1:1}}
51   \else
52     \geometry{top=2cm, bottom=2cm, width=35em,
53       headsep=14pt,hmarginratio={4:3}}
54   \fi
55 }{
56   \ifthenelse{\equal{\hwa@design}{v1}}{
57     \empty
58   }{
59     \ClassError{HomeworkAssignment}{Value '\hwa@design' for key 'design'
60       is not known}{The option design takes an argument to set the
61       Pagestyle to the one of a previous version. Acceptable values are
62       'v1', or 'v2'}
63   }
64 }

```

8.2 Translations

Load translations, currently supports English and German, Fallback is German

```

65 \DeclareTranslationFallback{aufgabe}{Aufgabe}
66 \DeclareTranslationFallback{loesung}{L"osung}
67 \DeclareTranslationFallback{beweis}{Beweis}
68 \DeclareTranslationFallback{uebungsgruppe}{\ "Ubungsgruppe}
69 \DeclareTranslationFallback{abgabe}{Abgabe}
70 \DeclareTranslationFallback{zuZeigen}{Zu zeigen}
71 \DeclareTranslationFallback{gegeben}{Gegeben}
72 \DeclareTranslationFallback{falls}{falls}
73 \DeclareTranslationFallback{Falls}{Falls}
74
75 \DeclareTranslation{German}{aufgabe}{Aufgabe}
76 \DeclareTranslation{German}{loesung}{L"osung}
77 \DeclareTranslation{German}{beweis}{Beweis}
78 \DeclareTranslation{German}{uebungsgruppe}{\ "Ubungsgruppe}

```

```

79 \DeclareTranslation{German}{abgabe}{Abgabe}
80 \DeclareTranslation{German}{zuZeigen}{Zu zeigen}
81 \DeclareTranslation{German}{gegeben}{Gegeben}
82 \DeclareTranslation{German}{falls}{falls}
83 \DeclareTranslation{German}{Falls}{Falls}
84
85 \DeclareTranslation{English}{aufgabe}{Problem}
86 \DeclareTranslation{English}{loesung}{Solution}
87 \DeclareTranslation{English}{beweis}{Proof}
88 \DeclareTranslation{English}{uebungsgruppe}{Tutorial}
89 \DeclareTranslation{English}{abgabe}{Deadline}
90 \DeclareTranslation{English}{zuZeigen}{To show}
91 \DeclareTranslation{English}{gegeben}{Given}
92 \DeclareTranslation{English}{falls}{if}
93 \DeclareTranslation{English}{Falls}{If}

```

8.3 Headers & Footers

Sets the page-headers.

All headers are cleread before they get any Text — just to be sure.

The headers look like specified above (4.1). Also inserts the Titlepage.

```

94
95 \fancypagestyle{firstpage}{
96   %
97   \fancyhf{}
98   % clear all six fields
99   \renewcommand{\headrulewidth}{.7pt}
100  \renewcommand{\footrulewidth}{0pt}
101  \fancyfoot[R]{\thepage}
102  \fancyhead[L]{\hwa@tutorium}
103  \fancyhead[R]{\@date }
104  \fancypagestyle{followingpage}{
105    %
106    \fancyhf{}
107
108    \ifthenelse{\equal{\hwa@design}{v2}}{
109      \ifhwa@twoside % IF
110
111      \fancyhead[RO]{\@author}
112      \fancyhead[LO]{\hwa@kurs\
113        \hwa@tutorium}
114      \fancyhead[LE]{\GetTranslation{abgabe}: \hwa@abgabe}
115      \fancyfoot[RO,LE]{\thepage}
116
117      \else %ELSE
118
119      \fancyhead[R]{\hwa@kurs\
120        \@author}
121      \fancyhead[L]{\hwa@tutorium\

```

```

122     \GetTranslation{abgabe}: \hwa@abgabe}
123     \fancyfoot[R]{\thepage}
124     \fi %ENDIF
125   }{
126     \ifthenelse{\equal{\hwa@design}{v1}}{
127       \fancyhead[RE,LO]{\@author}
128       \fancyhead[LE,RO]{\hwa@kurs\
129         \GetTranslation{abgabe}: \hwa@abgabe}
130       \fancyfoot[RE,LO]{\thepage}
131     }{
132       \ClassError{HomeworkAssignment}{Value '\hwa@design' for key 'design'
133         is not known}{The option design takes an argument to set the
134         Pagestyle to the one of a previous version. Acceptable values are
135         'v1', or 'v2'}
136     }
137   }
138
139
140   \renewcommand{\headrulewidth}{0.7pt}
141   \renewcommand{\footrulewidth}{0pt} } \pagestyle{followingpage}
142 \AtBeginDocument{ \thispagestyle{firstpage}
143   \setlength{\headheight}{25pt} }

```

8.4 Internal commands

8.4.1 Counter-Commands

Counter--Commands These are used to output the Exercise numbers in the desired style

```

144 \newcommand{\hwa@problemno}{\arabic{problem}}
145 \newcommand{\hwa@subproblemno}{\alph{subproblem}}
146 \newcommand{\hwa@subsubproblemno}{\roman{subsubproblem}}

```

8.4.2 Counter-Style Parser

Counter--Style Parser This takes a style-input (#1), one of the three previous defined commands (#2) and the corresponding counter (#3) to redefine #1, so that it corresponds to #2. See 8.4.3 for example usement.

```

147 \newcommand{\hwa@parseCounterStyle}[3]{
148   \ifthenelse{\equal{#1}{arabic}}{ \renewcommand{#2}{\arabic{#3}} }{
149     \ifthenelse{\equal{#1}{roman}}{ \renewcommand{#2}{\roman{#3}} }{
150       \ifthenelse{\equal{#1}{alph}}{ \renewcommand{#2}{\alph{#3}} }{
151         \ifthenelse{\equal{#1}{Alph}}{ \renewcommand{#2}{\Alph{#3}} }{
152           \ifthenelse{\equal{#1}{Roman}}{
153             \renewcommand{#2}{\Roman{#3}} }{
154               \ClassError{HomeworkAssignment}{Invalid Value #1 for
155                 option Counter-Styling}{Possible Values are alph,
156                 arabic, Arabic, roman or Roman.} } } } } }

```


8.4.3 Counter-Commands II

ounter--Style ParserCommands II Redefines the three counter-commands

```

157 \hwa@parseCounterStyle{\hwa@problemsty}{\hwa@problemno}{problem}
158 \hwa@parseCounterStyle{\hwa@subproblemsty}{\hwa@subproblemno}{subproblem}
159 \hwa@parseCounterStyle{\hwa@subsubproblemsty}{\hwa@subsubproblemno}{subsubproblem}

```

8.5 Commands

```

\subject Defines \kurs. \subject equals \kurs
160 \newcommand{\hwa@kurs}{?\GetTranslation{subject}??}
161 \newcommand{\subject}[1]{\renewcommand{\hwa@kurs}{#1}}
162 \newcommand{\kurs}[1]{\subject{#1}}

\tutorial Defines \tutorial. \tutorial equals \tutorial
163 \newcommand{\hwa@tutorial}{?\GetTranslation{uebungsgruppe}??}
164 \newcommand{\tutorial}[1]{\renewcommand{\hwa@tutorial}{#1}}
165 \newcommand{\tutorial}[1]{\tutorial{#1}}

\deadline Defines \deadline. \abgabe equals \deadline
166 \newcommand{\hwa@abgabe}{\today}
167 \newcommand{\deadline}[1]{\def\hwa@abgabe{#1}}
168 \newcommand{\abgabe}[1]{\deadline{#1}}

\maketitle Overrides maketitle.
169
170 \renewcommand{\maketitle} {
171   \twocolumn[{\%
172     \begin{centering}
173       \huge{\textbf{\hwa@kurs}} \vspace{.25cm} {\hrule height 2pt}
174       \vspace{.25cm} \large
175       \GetTranslation{abgabe}: \hwa@abgabe\\
176       \vspace{.5cm} \hrule \vspace{.25cm}
177       \normalsize{\@author}\\
178       \vspace{.25cm} \hrule \vspace{.25cm} \normalsize
179     \end{centering}
180   }]
181 }

Defines and initialize all counters.
182 \newcounter{problem} \setcounter{problem}{0}
183 \newcounter{subproblem}[problem] \setcounter{subproblem}{0}
184 \newcounter{subsubproblem}[subproblem] \setcounter{subsubproblem}{0}
185

Defines ‘plain’ sectioning-commands. See 3.2 for more informations.
186 \newcommand{\problem}[1]{\@startsection{problem}%Name
187   {1}%Level
188   {\z@}%indent

```

```

189 {-2em \@plus -1em \@minus -1em}%beforeskip
190 {1ex \@plus .5ex}%afterskip
191 {\normalfont\Large\bfseries}%style
192 *{#1} \addcontentsline{toc}{section}{#1} }
193
194 \newcommand{\subproblem}[1]{\@startsection{subproblem}%Name
195 {2}%Level
196 {\z@}%indent
197 {-1em \@plus -.5em \@minus -.5em}%beforeskip
198 {.5ex \@plus .5ex}%afterskip
199 {\normalfont\large\bfseries}%style
200 *{#1} \addcontentsline{toc}{subsection}{#1} }
201
202 \newcommand{\subsubproblem}[1]{\@startsection{subsubproblem}%Name
203 {3}%Level
204 {\z@}%indent
205 {-.5em}%beforeskip
206 {.5em}%afterskip
207 {\normalfont\bfseries}%style
208 *{#1} }
209
210 \newcommand{\solution}[1] [] {\@startsection{solution}%Name
211 {4}%Level
212 {\parindent}%indent
213 {-.1em}%beforeskip
214 {\z@}%afterskip
215 {\normalfont\bfseries}%style
216 *{\GetTranslation{loesung}\ifthenelse{\equal{#1}{}} {} { #1}:~~ } }
217
218 \newcommand{\proof}[1] [] {\@startsection{proof}%Name
219 {4}%Level
220 {\parindent}%indent
221 {-.1em}%beforeskip
222 {\z@}%afterskip
223 {\normalfont\bfseries}%style
224 *{\GetTranslation{beweis}\ifthenelse{\equal{#1} {} } {} { #1}:~~ } }
225
226 \newcommand{\toShow}[1] [] {\@startsection{to show}%Name
227 {4}%Level
228 {\parindent}%indent
229 {-.1em}%beforeskip
230 {\z@}%afterskip
231 {\normalfont\bfseries}%style
232 *{\GetTranslation{zuZeigen}\ifthenelse{\equal{#1} {} } {} { #1}:~~ } }
233
234 \newcommand{\given}[1] [] {\@startsection{given}%Name
235 {4}%Level
236 {\parindent}%indent
237 {-.1em}%beforeskip
238 {\z@}%afterskip

```

```

239 {\normalfont\bfseries}%style
240 *{\GetTranslation{gegeben}\ifthenelse{\equal{#1} {} } {} { #1:~ } }
241
    Defines ‘better’ sectioning commands. See 3.2 and 3.2.2 for more informations.
242 \newcommand{\newproblem}[1][\stepcounter{problem}
243 \ifthenelse{\equal{#1}{} } {} {\setcounter{problem}{#1}}
244 \problem{\GetTranslation{aufgabe} \hwa@problemno} }
245
246 \newcommand{\newsubproblem}[1][\stepcounter{subproblem}
247 \ifthenelse{\equal{#1}{} } {} {\setcounter{subproblem}{#1}}
248 \subproblem{\GetTranslation{aufgabe} \hwa@problemno}.\hwa@subproblemno} }
249
250 \newcommand{\newsbsubproblem}[1][\stepcounter{subsubproblem}
251 \ifthenelse{\equal{#1}{} } {} {\setcounter{subsubproblem}{#1}}
252 \subsubproblem{\hwa@subsubproblemno}} }
253

```

End of Proof

```

254 \newcommand{\QED}{\begin{flushright}
255 \textit{QED}
256 \end{flushright}
257 }
258 \newcommand{\EOP}{\begin{flushright}
259 $\square$
260 \end{flushright}
261 }
262 \newcommand{\eop}{\hfill$\blacksquare$}

```

c demonstrandum at iucundum est

```

263 \newcommand{\QED}{\begin{flushright}
264 $\triangle$
265 \end{flushright}
266 }
267 \newcommand{\qed}{\hfill$\triangle$}

```

The following Macros are all stolen (and adapted) from occloxiun (see 3.4)

Math Common Set Symbols

```

268 \newcommand{\N}{\ensuremath{\mathbb{N}}}
269 \newcommand{\Z}{\ensuremath{\mathbb{Z}}}
270 \newcommand{\R}{\ensuremath{\mathbb{R}}}
271 \newcommand{\Q}{\ensuremath{\mathbb{Q}}}
272 \newcommand{\C}{\ensuremath{\mathbb{C}}}
273 \newcommand{\F}{\ensuremath{\mathbb{F}}}
274 % The last one is mine
275 \newcommand{\Primes}{\ensuremath{\mathbb{P}}}

```

Mathematical Functions

```

276 \newcommand{\GL}{\ensuremath{\text{GL}}}

```

```

277 \newcommand{\id}{\ensuremath{\text{id}}}
278 \newcommand{\diff}[1]{\ensuremath{\frac{d}{d\#1}}}
279 \newcommand{\dx}{\text{:dx}}

```

Mathematical Functions

```

280 \newcommand{\divides}{\ensuremath{\mid}}
281 \newcommand{\property}{\ensuremath{\mid}}
282
283 \newcommand{\Var}{\ensuremath{\text{Var}}}
284 \newcommand{\Perm}{\ensuremath{\text{Perm}}}
285 \newcommand{\MComb}{\ensuremath{\text{MComb}}}
286 \newcommand{\Comb}{\ensuremath{\text{Comb}}}
287
288 \renewcommand{\dim}[1][\text{cm}]{\ensuremath{\text{dim}_{\#1}}}
289 \renewcommand{\Im}{\ensuremath{\text{Im}}}
290 \newcommand{\modulo}[2]{\#1/\!\!\#2\text{:}}
291
292 \newcommand{\Pot}{\ensuremath{\text{Pot}}}
293 \newcommand{\Map}{\ensuremath{\text{Map}}}
294
295 \newcommand{\excup}{\ensuremath{\stackrel{\cdot}{\cup}}}
296
297 \newcommand{\falls}{\text{\ \GetTranslation{falls}}\ }
298
299 \newcommand{\Bin}{\ensuremath{\text{Bin}}}
300 \renewcommand{\char}{\ensuremath{\text{char}}}

```

Math Big Quantors

```

301 \let\forall\forall
302 \let\exists\exists
303 \renewcommand{\forall}{\hspace{2pt}\forall\hspace{2pt}}
304 \renewcommand{\exists}{\hspace{2pt}\exists\hspace{2pt}}
305 \newcommand{\bigforall}{\mbox{\raisebox{-2pt}{\Large $\mathsurround{4pt}\forall$}}}
306 \newcommand{\bigexists}{\mbox{\raisebox{-2pt}{\Large $\mathsurround{4pt}\exists$}}}

```

The End

```

307 \endinput

```