

# The homeworkassignment\*class<sup>†</sup>

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

November 18, 2017

## Contents

<b>1</b>	<b>Abstract</b>	<b>2</b>
<b>2</b>	<b>Dependencies</b>	<b>2</b>
2.1	Mandatory Dependencies . . . . .	2
2.2	Recommended Dependencies . . . . .	3
<b>3</b>	<b>Options</b>	<b>4</b>
<b>4</b>	<b>Page-Layout</b>	<b>6</b>
4.1	Headers & Footers . . . . .	6
4.2	Enhance Mathenvironments . . . . .	7
<b>5</b>	<b>Commands</b>	<b>8</b>
5.1	Counters and Constants . . . . .	8
5.2	Document Informations . . . . .	8
5.2.1	Inherited from <code>article</code> . . . . .	9
5.3	Sectioning . . . . .	9
5.3.1	‘plain’ Sectioning . . . . .	9
5.3.2	‘better’ Sectioning . . . . .	10
5.4	Useful Macros . . . . .	12
5.4.1	QUOD ERAT DEMUNSTARNDUM, End of Proof . . . . .	12
5.4.2	QUOD NON ERAT DEMUNSTARNDUM AT IUCUNDUM EST . . . . .	12
5.4.3	Stolen Goods . . . . .	12
5.4.4	Rounding . . . . .	14
5.5	Grading Table . . . . .	14
<b>6</b>	<b>Development and support</b>	<b>15</b>

---

\*The name was changed with version v3.0, to become compatible with CTANs guidelines and to maintain a degree of backwards compatibility. The class was called `HomeworkAssignment` prior to v3.0

<sup>†</sup>This document corresponds to `homeworkassignment` v2.5f, dated 2017/11/18.

<b>7</b>	<b>Changelog</b>	<b>15</b>
7.1	Version-Scheme . . . . .	16
<b>8</b>	<b>Examples</b>	<b>17</b>
<b>9</b>	<b>Translations</b>	<b>17</b>
<b>10</b>	<b>Implementation</b>	<b>18</b>
10.1	Packages & Options . . . . .	18
10.2	TikZ-Styles . . . . .	18
10.3	Geometry . . . . .	18
10.4	Internal commands . . . . .	18
10.4.1	Counter-Commands . . . . .	18
10.4.2	Counter-Style Parser . . . . .	18
10.4.3	Counter-Commands II . . . . .	19
10.4.4	Grading-table . . . . .	19
10.5	Commands . . . . .	20

## 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 Dependencies

### 2.1 Mandatory Dependencies

This class is build uppon article, so of course the first dependency is:

`article` 1992 LESLIE LAMPORT, 1994-97 FRANK MITTELBACH JOHANNES BRAAMS, THE L<sup>A</sup>T<sub>E</sub>X-TEAM, <https://www.ctan.org/pkg/kvoptions>,

Because I am very lazy, the `homeworkassignment` is “a little bit” bloated. These are all required packages:

`kvoptions` HEIKO OBERDIEK, <https://www.ctan.org/pkg/kvoptions>,  
for `key=value`-style options

`suffix` DAVID KASTRUP, <https://www.ctan.org/pkg/suffix>,  
Makes it easy to define `\macro*` commands

`xifthen` JOSSELIN NOIREL, <https://www.ctan.org/pkg/xifthen>,  
For if-else-structures

`translations` CLEMENS NIEDERBERGER, <https://www.ctan.org/pkg/translations>,  
Implements an easy method of translations.

**amsmath** THE L<sup>A</sup>T<sub>E</sub>X-TEAM, FRANK MITTELBACH RAINER SCHÖPF, ET AL.,  
<https://www.ctan.org/pkg/amsmath>,  
 For better math-typesetting

**amssymb** AMERICAN MATHEMATICAL SOCIETY, [mirror.ctan.org/fonts/amsfonts/doc/amssymb.pdf](https://www.ctan.org/mirror.ctan.org/fonts/amsfonts/doc/amssymb.pdf),  
 For more mathematical symbols

**etoolbox** PHILIPP LEHMAN (INACTIVE), JOSEPH WRIGHT, <https://www.ctan.org/pkg/etoolbox>,  
 The package is a toolbox of programming facilities geared primarily towards L<sup>A</sup>T<sub>E</sub>Xclass and package authors

**array** FRANK MITTELBACH, DAVID CARLISLE, THE L<sup>A</sup>T<sub>E</sub>X-TEAM, <https://www.ctan.org/pkg/array>,  
 A new implementations for tables and arrays

**xparse** FRANK MITTELBACH, CHRIS ROWLEY, DAVID CARLISLE, THE L<sup>A</sup>T<sub>E</sub>X3 PROJECT, <https://ctan.org/pkg/xparse>,  
 The package provides a high-level interface for producing documentlevel commands. In that way, it offers a replacement for L<sup>A</sup>T<sub>E</sub>X2<sub>ε</sub>'s `\newcommand` macro, with significantly improved functionality.

**gillius** BOB TENNENT, <https://ctan.org/pkg/gillius>,  
 A Gillian Sans inspired font, used for all sans serifs fonts

array possibly can be removed

## 2.2 Recommended Dependencies

These are not loaded automatically, but require a switch as option (see section 3). The switch is typically the name of the package.

**tikz** TILL TANTAU, MARK WIBROW, CHRISTIAN FEUERSÄNGER ET AL., <https://www.ctan.org/pkg/pgf>,  
 An incredible powerfull image tool. When loading TikZ, the homeworkassignment automatically loads a shipload of TikZ-librarys and own styles. See section 3 for more informations

**listings** CARSTEN HEINZ, BROOKS MOSES, JOBST HOFFMANN, <https://www.ctan.org/pkg/listings>,  
 For source-code. Sourcecode in the homeworkassignment is automatically framed, printed in `scriptsize`, and linebeals will be introduced

I intend to move these styles to a own package, so that they are usable without the homeworkassignment

Loads required Packages

```
1 \RequirePackage{suffix}
2 \RequirePackage{fancyhdr}
3 \RequirePackage{xifthen}
4 \RequirePackage{translations}
5 \PassOptionsToPackage{fleqn}{amsmath}
6 \RequirePackage{amsmath}
```

```

7 \RequirePackage{amssymb}
8 \RequirePackage{etoolbox}
9 \RequirePackage{array}
10 \RequirePackage{xparse}
11 \RequirePackage{gillius2}

```

### 3 Options

KV-Options is essential for this.

```

12 \RequirePackage{kvoptions}
13 \SetupKeyvalOptions{ family=hwa,
14   prefix=hwa@ }
15 \DeclareDefaultOption{\PassOptionsToClass{\CurrentOptionKey}{article}}

```

`problemstyle=<1>` These options allow the customization of the displayed numbers. For Example, if  
`subproblemstyle=<1>` `problemstyle=Roman`, `subproblemstyle=arabic`, `subsubproblemstyle=roman`  
`subsubproblemstyle=<1>` 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 `arabic`, `Alph`, `alph`, `Roman`, and `roman`. Standard values are:  
`problemstyle=arabic`, `subproblemstyle=alph`, `subsubproblemstyle=roman`.

```

16 \DeclareStringOption[arabic]{problemsty}
17 \DeclareStringOption[alph]{subproblemsty}
18 \DeclareStringOption[roman]{subsubproblemsty}

```

`tikz` Loads TikZ-Package and a couple of Styles, usefull for Papers in Computer-Science  
and mathematics. See 3 for more informations

```

19 \DeclareBoolOption[false]{tikz}

```

`listings` Loads Listings Package and sets listing-layout to use a small fontsize. Adds  
indication for linebreaks.

```

20 \DeclareBoolOption[false]{listings}

```

`oneside, twoside` Changes layout. `oneside` is the complementary option to `twoside`  
Standard layout is `twopaged`.

```

21 \DeclareBoolOption[true]{twoside}
22 \DeclareComplementaryOption{oneside}{twoside}

```

`onecolumn, twocolumn` Changes layout. `onecolumn` is the complementary option to `twocolumn`.  
Standard Layout has two columns

```

23 \DeclareBoolOption[true]{twocolumn}
24 \DeclareComplementaryOption{onecolumn}{twocolumn}

```

`hlines=<1>` Key-Value-option. Takes the level of hlines. Available are `all`, `decreased`, `header`,  
`none`, with decreasing number of lines; `none` displays none, `header` only the one  
under headers and `decreased` adds the big line in the title, while `all` displays all.

```

25 \DeclareStringOption[all]{hlines}

```

Loads article and processes the options

```

26 \ProcessKeyvalOptions*
27 \ifhwa@twoside

```

```

28 \PassOptionsToClass{twoside}{article}
29 \else
30 \PassOptionsToClass{oneside}{article}
31 \fi
32 \ifhwa@twocolumn
33 \PassOptionsToClass{twocolumn}{article}
34 \else
35 \PassOptionsToClass{onecolumn}{article}
36 \fi
37 \ifhwa@listings
38 \RequirePackage{listings}
39 \lstset{
40   frame = single,
41   breaklines = true,
42   postbreak=\raisebox{0ex}[0ex][0ex]{\ensuremath{\hookrightarrow}\space}},
43   basicstyle=\scriptsize
44 }
45 \else
46 \empty
47 \fi
48 \LoadClass{article}

```

`\hwa@hline@L...` Defines new commands to output desired lines and change the constant `\hwa@headrulewidth`

ATTENTION: `\hwa@hline@LONE` breaks the line automatically, in opposite to `\hwa@hline@LTWO`

```

49
50 \newcommand{\hwa@hline@LONE}{\vspace{.25cm} {\hrule height 2pt}
51   \vspace{.25cm}}
52 \newcommand{\hwa@hline@LTWO}{\vspace{.5cm} \hrule \vspace{.25cm}}
53 \newcommand{\hwa@headrulewidth}{.7pt}
54 \ifthenelse{equal{\hwa@hlines}{all}}{
55   \renewcommand{\hwa@hline@LONE}{\vspace{.25cm} {\hrule height 2pt}
56     \vspace{.25cm}}
57   \renewcommand{\hwa@headrulewidth}{.7pt}
58   \renewcommand{\hwa@hline@LTWO}{\vspace{.5cm} \hrule \vspace{.25cm}}
59 }{
60   \ifthenelse{equal{\hwa@hlines}{decreased}}{
61     \renewcommand{\hwa@hline@LONE}{\vspace{.25cm} {\hrule height 2pt}
62       \vspace{.25cm}}
63     \renewcommand{\hwa@headrulewidth}{.7pt}
64   }{\ifthenelse{equal{\hwa@hlines}{header}}{
65     \renewcommand{\hwa@headrulewidth}{.7pt}
66   }{\ifthenelse{equal{\hwa@hlines}{none}}{
67     \renewcommand{\hwa@headrulewidth}{0pt}
68   }{
69     \ClassError{homeworkassignment}{Value '\hwa@lines' for key 'hlines'

```

```

70         is not known}{The option hlines takes an argument to set which
71         hlines are drawn. Possible values are 'all','decreased' , 'header', and
72         'none'. 'all' is standard.}
73     }
74 }
75 \renewcommand{\hwa@hline@LONE}{~\\vspace{.5cm}}
76 }
77 \renewcommand{\hwa@hline@LTWO}{\vspace{.75cm}}
78 }

```

If tikz is Wanted, load Usefull Styles

```

79 \ifhwa@tikz
80 \RequirePackage{tikz}
81 \usetikzlibrary{shapes,arrows,positioning,decorations,
82     automata,backgrounds,petri,bending,
83     shapes.multipart}
84 \tikzset{
85     treenode/.style = {shape=circle, rounded corners,
86         draw, align=center},
87     graynode/.style = {fill=gray},
88     normalnode/.style = {treenode, font=\Large, bottom color=white},
89     array/.style = {rectangle split,
90         rectangle split horizontal,
91         rectangle split,
92         draw}
93 }
94 \fi

```

Make sure that this is the last Package loaded

```

95 \RequirePackage{geometry}
96 \ifhwa@twocolumn
97 \geometry{top=2cm, bottom=2cm, left=2cm,
98     headsep=14pt,hmarginratio={1:1}}
99 \else
100 \geometry{top=2cm, bottom=2cm, width=35em,
101     headsep=14pt,hmarginratio={4:3}}
102 \fi

```

## 4 Page-Layout

Initially, the homeworkassignment had a verry *special* appereance, which became much mor customizable with version 3.0, see ?? if you want to know how.

### 4.1 Headers & Footers

Sets the page-headers.

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

The headers have the date on the subject and the author on the right side, the tutorial, sheet-title and deadline on the left side, the pagenumber is displayed in the right footer.

If the document is twopaged, the informations in the headers are splittet, so that author and subject are displayed only on odd pages and the title on even, the pagenumber is displayed on the right side on odd pages and on the left side on even pages.

On the first page, only the date and tutorial will be displayed in the header, the rest of infomration should be in the title.

```

103 \fancypagestyle{firstpage}{
104   %
105   \fancyhf{}
106   % clear all six fields
107   \renewcommand{\headrulewidth}{\hwa@headrulewidth}
108   \renewcommand{\footrulewidth}{0pt}
109   \fancyfoot[R]{\thepage}
110   \fancyhead[L]{\hwa@tutorial}
111   \fancyhead[R]{\@date } }
112 \fancypagestyle{followingpage}{
113   \fancyhf{}
114   \ifhwa@twoside % IF
115   \fancyhead[R0]{\@author}
116   \fancyhead[L0]{\hwa@kurs\
117     \hwa@tutorial}
118   \fancyhead[LE]{
119     \ifthenelse{\equal{\hwa@sheetTitle}{}}{\hwa@sheetTitle\\}
120     \GetTranslation{abgabe}: \hwa@abgabe
121   }
122   \fancyfoot[R0,LE]{\thepage}
123
124   \else %ELSE
125
126   \fancyhead[R]{\hwa@kurs\
127     \@author}
128   \fancyhead[L]{\hwa@tutorial\
129     \ifthenelse{\equal{\hwa@sheetTitle}{}}{\hwa@sheetTitle\\}
130     \GetTranslation{abgabe}: \hwa@abgabe}
131   \fancyfoot[R]{\thepage}
132   \fi %ENDIF
133   \renewcommand{\headrulewidth}{\hwa@headrulewidth}
134   \renewcommand{\footrulewidth}{0pt}
135 }
136 \pagestyle{followingpage}

```

## 4.2 Enhance Mathenvironments

A couple of thigns, to make math-environments more beautifull and compact.

`\theequation` Displays equation-numbers as upper-case roman numbers.  
137 `\renewcommand{\theequation}{\Roman{equation}}`

`\allowdisplaybreaks` Allow pagebreaks in Mathmode  
138 `\allowdisplaybreaks`

## 5 Commands

### 5.1 Counters and Constants

Defines some constants

`\hwa@pointboxsize` Explains it self.  
139 `\newcommand{\hwa@pointboxsize}{3em}`

### 5.2 Document Informations

`\subject`, `\kurs` Sets the subject of the document. Takes the subject as argument. Standard Value is “Kein Kurs”  
`\kurs` is deprecated.

```
140 \newcommand{\hwa@kurs}{?\GetTranslation{subject}??} % To store the value
141 \newcommand{\subject}[1]{\renewcommand{\hwa@kurs}{#1}}
142 \newcommand{\kurs}[1]{\subject{#1}}
```

`\tutorial`, `\tutorium` 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.

```
143 \newcommand{\hwa@tutorium}{?\GetTranslation{uebungsgruppe}??} % To store the value
144 \newcommand{\tutorial}[1]{\renewcommand{\hwa@tutorium}{#1}}
145 \newcommand{\tutorium}[1]{\tutorial{#1}}
```

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

```
146 \newcommand{\hwa@abgabe}{\today} % To store the value
147 \newcommand{\deadline}[1]{\def\hwa@abgabe{#1}}
148 \newcommand{\abgabe}[1]{\deadline{#1}}
```

`\sheetTitle` Sets a descriptonal Title of the Sheet, will be written in the header of every page.  
149 `\newcommand{\hwa@sheetTitle}{}
150 \newcommand{\sheetTitle}[1]{\def\hwa@sheetTitle{#1}}`



### 5.2.1 Inherited from article

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

## 5.3 Sectioning

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

### 5.3.1 ‘plain’ Sectioning

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

```
151 \DeclareDocumentCommand\problem{m o}{\@startsection{problem}%Name
152   {1}%Level
153   {\z@}%indent
154   {-2em \@plus -1em \@minus -1em}%beforeskip
155   {1ex \@plus .5ex}%afterskip
156   {\normalfont\Large\sffamily\bfseries}%style
157   *{#1
158     \IfNoValueF{#2}{
159       \hfill
160       \frame{\framebox[\hwa@pointboxsize]{
161         \hfill \normalfont{\large/\small{#2}}}}
162     }
163   }
164   \addcontentsline{toc}{section}{#1}
165 }
166
167 \DeclareDocumentCommand\subproblem{m o}{\@startsection{subproblem}%Name
168   {2}%Level
169   {\z@}%indent
170   {-1em \@plus -.5em \@minus -.5em}%beforeskip
171   {.5ex \@plus .5ex}%afterskip
172   {\normalfont\large\sffamily\bfseries}%style
173   *{#1
174     \IfNoValueF{#2}{
175       \hfill \framebox[\hwa@pointboxsize]{
176         \hfill \normalfont\large/\small{#2}}
177     }
178   }
179   \addcontentsline{toc}{subsection}{#1}
180 }
181
```

```

182 \DeclareDocumentCommand\subsubproblem{m o}{\@startsection{subsubproblem}%Name
183   {3}%Level
184   {\z@}%indent
185   {- .5em}%beforeskip
186   {.5em}%afterskip
187   {\normalfont \sffamily\bfseries}%style
188   *{#1
189     \IfNoValueF{#2}{
190       \hfill \framebox[\hwa@pointboxsize]{
191         \hfill\normalfont\large/\scriptsize{#2}}
192     }
193   }
194 }
195

```

**\keyword** Creates a new Paragraph ,which will start with the Argument in Bold, followed by two non-breaking spaces.

```

196 \newcommand{\keyword}[1]{\@startsection{keyword}%Name
197   {4}%Level
198   {\parindent}%indent
199   {- .1em}%beforeskip
200   {\z@}%afterskip
201   {\normalfont \sffamily\bfseries}%style
202   *{#1~}
203 }

```

The following Macros make use of **\keyword**, so it is suggested to use them instead.

<b>\solution</b> <b>\proof</b> <b>\given</b> <b>\toShow</b> <b>\assumption</b> <b>\supposeThat</b>	They work like <b>\keyword</b> , but take only an optional Argument print out “Solution”, “Proof” “Given”, “To show”, “Assumption”, and “Suppose that”, respectively <sup>1</sup> , via <b>\keyword</b> . If an argument is passed, they print out this argument after the keyword. They are not mentioned in the table of contents.	<pre> 204 \newcommand{\solution}[1][\keyword{\GetTranslation{loesung}}\ifstrempy{#1}{~#1:}} 205 \newcommand{\proof}[1][\keyword{\GetTranslation{beweis}}\ifstrempy{#1}{~#1:}} 206 \newcommand{\toShow}[1][\keyword{\GetTranslation{zuZeigen}}\ifstrempy{#1}{~#1:}} 207 \newcommand{\given}[1][\keyword{\GetTranslation{gegeben}}\ifstrempy{#1}{~#1:}} 208 \newcommand{\assumption}[1][\keyword{\GetTranslation{Annahme}}\ifstrempy{#1}{~#1:}} 209 \newcommand{\supposeThat}[1][\keyword{\GetTranslation{Angenommen-dass}}\ifstrempy{#1}{~#1:}} </pre>
---	--	---

### 5.3.2 ‘better’ Sectioning

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

<b>\newproblem</b> <b>\newproblem*</b> <b>\newsproblem</b> <b>\newsproblem</b>	These commands require no argument, and automatically create a numbered ti- <hr style="width: 20%; margin-left: 0;"/> <sup>1</sup> As of v1.6, Translations are added, depending on the choosen Language, there may be an other Text displayed. See section 9 for all Translations
---	--

tle. They have two optional arguments: `\newproblem[#1]{#2}` where #1 is the (sub(sub))problem-number and #2 are the points. If there is a number of Points assigned to a (sub(sub))problem, then the command will generate a box to write the reched number of points down next to it.

Normally, `\newproblem` adds the new Created Problem to the grading-table (see subsection 5.5), `\newproblem*` does not do this.

They use coutners, of course:

```

210 \newcounter{problem} \setcounter{problem}{0}
211 \newcounter{subproblem}[problem] \setcounter{subproblem}{0}
212 \newcounter{subsubproblem}[subproblem] \setcounter{subsubproblem}{0}
213
214 \DeclareDocumentCommand\newproblem{0{ } g}{
215   \IfNoValueTF{#2}{
216     \newproblem*{#1}
217     \addtoGradingTable{\# \hwa@problemno}
218   }{
219     \IfNoValueF{#1}{
220       \stepcounter{problem}% to reset the lower counters
221       \setcounter{problem}{#1}
222     }
223     \problem{\GetTranslation{aufgabe} \hwa@problemno}[#2]
224     \addtoGradingTable{\# \hwa@problemno}{/#2}
225   }
226 }
227
228 \WithSuffix\newcommand\newproblem*[1] [] {\stepcounter{problem}
229   \ifthenelse{\equal{#1}{}} { } {
230     \stepcounter{problem}% to reset the lower counters
231     \setcounter{problem}{#1}
232   } \problem{\GetTranslation{aufgabe} \hwa@problemno}
233 }
234
235 \DeclareDocumentCommand\newsubproblem{0{ } g}{
236   \stepcounter{subproblem}
237   \ifthenelse{\equal{#1}{}} { } {
238     \setcounter{subproblem}{#1}
239   } \IfNoValueTF{#2}{
240     \subproblem{\GetTranslation{aufgabe}
241       \hwa@problemno}{.\hwa@subproblemno}
242   }
243   {
244     \subproblem{\GetTranslation{aufgabe}
245       \hwa@problemno}{.\hwa@subproblemno}[#2]
246   }
247 }
248
```

```

249 \DeclareDocumentCommand\newsussubproblem{0{ } g}{
250   \stepcounter{sussubproblem}
251   \ifthenelse{\equal{#1}{}}{ } { \setcounter{sussubproblem}{#1}}
252   \IfNoValueTF{#2}{
253     \sussubproblem{\hwa@sussubproblemno}}
254   }
255   {
256     \sussubproblem{\hwa@sussubproblemno}}[#2]
257   }
258 }
259

```

## 5.4 Useful Macros

### 5.4.1 QUOD ERAT DEMUNSTARNDUM, End of Proof

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

```

260 \newcommand{\QED}{\begin{flushright}
261   \textsc{Qed}
262 \end{flushright}}
263 }
264 \newcommand{\EOP}{\begin{flushright}
265   \(\square\)
266 \end{flushright}}
267 }
268 \newcommand{\eop}{\hfill\(\blacksquare\)}

```

### 5.4.2 QUOD NON ERAT DEMUNSTARNDUM AT IUCUNDUM EST

`\QNEd`  
`\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 or things wich are not proofed mathematically,  
but are explained in a ay, whcih lets no doubt on their correctness.

```

269 \newcommand{\QNEd}{\begin{flushright} \(\triangle\)
270 \end{flushright}}
271 }
272 \newcommand{\qned}{\hfill\(\triangle\)}

```

### 5.4.3 Stolen Goods

»Das ist alles nur geklaut«

~Tobias Künzel

These Commands are not mine, they are all stolen from Alexander Bartolomey's<sup>2</sup> `amath-Class`<sup>3</sup>

<code>\N</code>			
<code>\Z</code>	Defines a set of mathematical sets, which are very usefull (see Table 1)		
<code>\R</code>			
<code>\Q</code>	Command	Output	Description
<code>\C</code>	<code>\N</code>	$\mathbb{N}$	Natural Numbers
<code>\F</code>	<code>\Z</code>	$\mathbb{Z}$	Whole Numbers
<code>\Primes</code>	<code>\Q</code>	$\mathbb{Q}$	Rational Numbers
	<code>\R</code>	$\mathbb{R}$	Real Numbers
	<code>\C</code>	$\mathbb{C}$	Complex Numbers
	<code>\F_n</code>	$\mathbb{F}_n$	Prime Field to base $n$
	<code>\Primes</code> <sup>4</sup>	$\mathbb{P}$	Set of all Primes

Table 1: Field-Commands

```

273 \newcommand{\N}{\ensuremath{\mathbb{N}}}
274 \newcommand{\Z}{\ensuremath{\mathbb{Z}}}
275 \newcommand{\R}{\ensuremath{\mathbb{R}}}
276 \newcommand{\Q}{\ensuremath{\mathbb{Q}}}
277 \newcommand{\C}{\ensuremath{\mathbb{C}}}
278 \newcommand{\F}{\ensuremath{\mathbb{F}}}
279 % The last one is mine
280 \newcommand{\Primes}{\ensuremath{\mathbb{P}}}
```

**Functions and Operators** Output usefull Plaintext-Operators and Functions. See table 2. Require Math-mode

	Command	Output
<code>\divides</code>	and property	Prints a vertical Bar
	<code>\Var</code>	Var
	<code>\Perm</code>	Perm
	<code>\Comb</code>	Comb
	<code>\MComb</code>	MComb
	<code>\Pot</code>	Pot
	<code>\Map</code>	Map
	<code>\Bin</code>	Bin
	<code>\GL</code>	GL
	<code>\id</code>	id
	<code>\dx</code>	$dx$
	<code>\excup</code>	$\dot{\cup}$

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

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

<sup>4</sup>Has to be `\Primes`, because `\P` is already in use

$$\backslash\mathrm{diff}\{<1>\} \quad \frac{d}{d<1>}$$

Table 2: Text-like Functions

`\falls` prints out »falls«<sup>5</sup>

#### 5.4.4 Rounding

Require Mathmode

Command	Output	Meaning
<code>\floor{&lt;1&gt;}</code>	$\lfloor <1> \rfloor$	floor $<1>$
<code>\ceil{&lt;1&gt;}</code>	$\lceil <1> \rceil$	ceil $<1>$
<code>\roundHU{&lt;1&gt;}</code>	$\lfloor <1> \rfloor$	Round $<1>$ “half up” ( $\lfloor <1> + \frac{1}{2} \rfloor$ )
<code>\roundHD{&lt;1&gt;}</code>	$\lfloor <1> \rfloor$	Round $<1>$ “half down” ( $-\lfloor <1> - \frac{1}{2} \rfloor$ )

Table 3: Rounding Functions

### 5.5 Grading Table

This Document-Class is still mainly designed for Homework, so it would be nice, if there was a table to write Grades into, wouldn’t it?

<code>\addToGradingTable</code>	Adds the given parameter as an exercise to the Grading-Table. All Problems, created with <code>\newproblem</code> are added automatically.
<code>\makeGradingTable</code>	Prints out the Table containig all Defined exercises ( $\neq$ Problems). Like <code>\tableofcontent</code> , it needs a second run of L <sup>A</sup> T <sub>E</sub> X, until all are added. See example documents fot output

---

<sup>5</sup>In German, actual Translation may differ

## 6 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.

## 7 Changelog

**v1.0 - 2016/10/23** Initial

**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 `\QED`

**v2.0 - 2017/05/23** “Layout 2.0”

    • Change Margins

    • Add Option to select older Page-Style

    • Change `standardlayout` to `twocolumn` and `twoside`

    • Steal Use Macros by Alexander Bartolomey (See 5.4.3)

    • Add some TikZ-Styles

    • Add round functions

**v2.2 - 2017/06/17**    • Add Grading-table

    • Add `\keyword`, `\assumption`, and `\supposeThat`

- Add `\newproblem*`
  - Add `\sheetTitle`
  - Change equation-numbering to uppercase roman
- v2.2.1 - 2017/06/20**    • Fix error with commands like `\solution` and `\keyword`.
- v2.4 - 2017/04/07**    • Fix math alignment
- Add option for flushed left equations
  - Update amath port to use
- v3.0 - pending** “WS 2017”
- Rename to `homeworkassignment`
  - Add Environment for various proofs
  - Add points for exercises and a place to fill them in
  - Add option to remove or decrease the hlines

## 7.1 Version–Scheme

Since Version 2.0 the following version–scheme applies:

**Major Version** has to be increased, if

- there are changes, which create visible changes in the output of existing documents (except for bugfixes), or
- a command is removed or changed in a way, that existing documents do not compile with the new version.

**Minor Version** has to be increased, if

- new backwards compatible commands are introduced
- Bugfixes may be introduced too.

The minor version of stable releases is always even, the minor version of development versions is always odd. (0 counts as even).

**Patches** May be introduced on Stable Branch. With every non-document-breaking bugfix, the patch–number has to be incremented.

Because Fixing Bugs is a part of development, development–versions do not have numeric patch–numbers, but alphabetic identifiers, directly after the minor–version.



## 8 Examples

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

## 9 Translations

Homeworkassignment currently supports English and German, fallback language is German. Unfortunately these two are the only Languages I am capable of translating reliably, so if you want to use an other language, I would be very happy if you would help me to translate homeworkassignment! Please open an issue, author a pull-request or send me an e-mail.

```
281 \DeclareTranslationFallback{aufgabe}{Aufgabe}
282 \DeclareTranslationFallback{loesung}{L\"osung}
283 \DeclareTranslationFallback{beweis}{Beweis}
284 \DeclareTranslationFallback{uebungsgruppe}{\"Ubungsgruppe}
285 \DeclareTranslationFallback{abgabe}{Abgabe}
286 \DeclareTranslationFallback{zuZeigen}{Zu zeigen}
287 \DeclareTranslationFallback{gegeben}{Gegeben}
288 \DeclareTranslationFallback{falls}{falls}
289 \DeclareTranslationFallback{Annahme}{Annahme}
290 \DeclareTranslationFallback{Angenommen-dass}{Angenommen, dass}
291
292 \DeclareTranslation{German}{aufgabe}{Aufgabe}
293 \DeclareTranslation{German}{loesung}{L\"osung}
294 \DeclareTranslation{German}{beweis}{Beweis}
295 \DeclareTranslation{German}{uebungsgruppe}{\"Ubungsgruppe}
296 \DeclareTranslation{German}{abgabe}{Abgabe}
297 \DeclareTranslation{German}{zuZeigen}{Zu zeigen}
298 \DeclareTranslation{German}{gegeben}{Gegeben}
299 \DeclareTranslation{German}{falls}{falls}
300 \DeclareTranslation{German}{Falls}{Falls}
301 \DeclareTranslation{German}{Annahme}{Annahme}
302 \DeclareTranslation{German}{Angenommen-dass}{Angenommen, dass}
303
304 \DeclareTranslation{English}{aufgabe}{Problem}
305 \DeclareTranslation{English}{loesung}{Solution}
306 \DeclareTranslation{English}{beweis}{Proof}
307 \DeclareTranslation{English}{uebungsgruppe}{Tutorial}
308 \DeclareTranslation{English}{abgabe}{Deadline}
309 \DeclareTranslation{English}{zuZeigen}{To show}
310 \DeclareTranslation{English}{gegeben}{Given}
311 \DeclareTranslation{English}{falls}{if}
312 \DeclareTranslation{English}{Falls}{If}
313 \DeclareTranslation{English}{Annahme}{Assumption}
314 \DeclareTranslation{English}{Angenommen-dass}{Suppose that}
```

## 10 Implementation

The following part is very boring, but I have not found a solution to create a .cls-file without including the implemetation into the document. Loads L<sup>A</sup>T<sub>E</sub>X2e and sets the Version Loads the `article`, which is the base-class.

### 10.1 Packages & Options

### 10.2 TikZ-Styles

### 10.3 Geometry

### 10.4 Internal commands

`\hwa@maketitletext` Prints out the title with author etc. Used to reduce code duplication for two- and onecolumn styles

```
315 \newcommand{\hwa@maketitletext}{
316   \begin{centering}
317     \huge{\textsf{\textbf{\hwa@kurs}}}\hwa@hline@LONE \large
318     \ifthenelse{\equal{\hwa@sheetTitle}{}}{\textsf{\hwa@sheetTitle}}{\}
319     \GetTranslation{abgabe}: \hwa@abgabe\
320     \hwa@hline@LTW0
321     \normalsize{\@author}\
322     \hwa@hline@LTW0 \normalsize
323   \end{centering}
324 }
```

#### 10.4.1 Counter-Commands

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

```
325 \newcommand{\hwa@problemno}{\arabic{problem}}
326 \newcommand{\hwa@subproblemno}{\alph{subproblem}}
327 \newcommand{\hwa@subsubproblemno}{\roman{subsubproblem}}
```

#### 10.4.2 Counter-Style Parser

`Counter-Style Parser` This takes a style-input (#1), one of the three previous defined commands (#2) and the coresponding counter (#3) to redefine #1, so that it corresponds to #2. See 10.4.3 for example usement.

```
328 \newcommand{\hwa@parseCounterStyle}[3]{
329   \ifthenelse{\equal{#1}{arabic}}{\renewcommand{#2}{\arabic{#3}} }{
330     \ifthenelse{\equal{#1}{roman}}{\renewcommand{#2}{\roman{#3}} }{
331       \ifthenelse{\equal{#1}{alph}}{\renewcommand{#2}{\alph{#3}} }{
332         \ifthenelse{\equal{#1}{Alph}}{\renewcommand{#2}{\Alph{#3}} }{
333           \ifthenelse{\equal{#1}{Roman}}{
334             \renewcommand{#2}{\Roman{#3}} }{
335             \ClassError{homeworkassignment}{Invalid Value #1 for
336               option Counter-Styling}{Possible Values are alph,
337               arabic, Arabic, roman or Roman.} } } } }
```

### 10.4.3 Counter-Commands II

Counter-Style ParserCommands II Redefines the three counter-commands

```
338 \hwa@parseCounterStyle{\hwa@problemsty}{\hwa@problemno}{problem}
339 \hwa@parseCounterStyle{\hwa@subproblemsty}{\hwa@subproblemno}{subproblem}
340 \hwa@parseCounterStyle{\hwa@subsubproblemsty}{\hwa@subsubproblemno}{subsubproblem}
```

### 10.4.4 Grading-table

\hwa@gradingtbl@... Defines macros whose contents will be written to the AUX-File and read in the next run, and the usable commands. The later will contain the information, but have to be defined (incase the aux-file does not exist)

```
341 \edef\hwa@gradingtbl@aux@defs{}
342 \newcommand{\hwa@gradingtbl@aux@lineOne}{}
343 \newcommand{\hwa@gradingtbl@aux@lineTwo}{}
344
345 \edef\hwa@gradingtbl@defs{}
346 \newcommand{\hwa@gradingtbl@lineOne}{}
347 \newcommand{\hwa@gradingtbl@lineTwo}{}

```

\addToGradingTable

```
348 \DeclareDocumentCommand\addToGradingTable{m g}{
349   \edef\hwa@gradingtbl@aux@defs{\hwa@gradingtbl@aux@defs|p{\hwa@pointboxsize}}
350   \edef\hwa@gradingtbl@aux@lineOne{\hwa@gradingtbl@aux@lineOne{#1} &}
351   \IfNoValueTF{#2}{
352     \edef\hwa@gradingtbl@aux@lineTwo{\hwa@gradingtbl@aux@lineTwo &}
353   }{
354     \edef\hwa@gradingtbl@aux@lineTwo{\hwa@gradingtbl@aux@lineTwo\vfill\hfill
355       {\string\small #2} &}
356   }
357 }
```

W rite to aux

```
358 \AtEndDocument{%
359   \immediate\write\@auxout{%
360     \gdef\string\hwa@gradingtbl@defs{\hwa@gradingtbl@aux@defs}
361   }
362   \immediate\write\@auxout{%
363     \gdef\string\hwa@gradingtbl@lineOne{\hwa@gradingtbl@aux@lineOne}%
364   }
365   \immediate\write\@auxout{%
366     \gdef\string\hwa@gradingtbl@lineTwo{\hwa@gradingtbl@aux@lineTwo}%
367   }
368 }
```

\makeGradingTable Outputs a table to fill in the reached Points. Only shows Problems created by \newproblem.

Displays the according number of maximum points for each problem, if specified.  
Displays the total number of maximum Problems, if given by Argument

**Arguments** [1]: *Optional*. The total number of points reachable.

```

369 \DeclareDocumentCommand\makeGradingTable{o}{
370   \begin{table}[hb]
371     \centering
372     \large
373     \expandafter\tabular\expandafter{\hwa@gradingtbl@defs ||p{\hwa@pointboxsize}||}\hline
374     \hwa@gradingtbl@lineOne   $\Sigma$          \\ \hline \small
375     \hwa@gradingtbl@lineTwo   \IfNoValueTF{#1}{~}{\vfill\hfill/#1}\vspace{.15cm}\\ \hline
376     \endtabular
377   \end{table}
378 }

```

## 10.5 Commands

`\deadline` Defines `\deadline`. `\abgabe equals \deadline`

`\maketitle` Overrides `\maketitle`.

```

379
380 \renewcommand{\maketitle} {
381   \thispagestyle{firstpage}
382   \ifhwa@twocolumn{
383     \twocolumn[{
384       \hwa@maketitletext
385     }]
386   }\else{
387     \hwa@maketitletext
388   }\fi
389 }

```

Defines and initialize all counters.

Defines ‘plain’ sectioning-commands. See 5.3 for more informations.

Defines ‘better’ sectioning commands. See 5.3 and 5.3.2 for more informations.

t demonstrandum at iucundum est

Rounding brakets

Round brakets

```

390 \newcommand{\floor}[1]{\ensuremath{\left\lfloor #1 \right\rfloor}}
391 \newcommand{\ceil}[1]{\ensuremath{\left\lceil #1 \right\rceil}}
392 \newcommand{\roundHU}[1]{\ensuremath{\left\lceil #1 \right\rfloor}}
393 \newcommand{\roundHD}[1]{\ensuremath{\left\lfloor #1 \right\rceil}}

```

Mathematical Functions

```

394 \DeclareMathOperator{\GL}{GL}
395 \DeclareMathOperator{\id}{id}
396 \DeclareMathOperator{\Var}{Var}
397 \DeclareMathOperator{\Perm}{Perm}
398 \DeclareMathOperator{\MComb}{MComb}
399 \DeclareMathOperator{\Comb}{Comb}

```

