

The homeworkassignment*class[†]

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

November 7, 2017

Contents

1	Abstract	2
2	Options	2
2.1	Inherited options	2
3	Commands	3
3.1	Document Informations	3
3.1.1	Inherited from <code>article</code>	3
3.2	Sectioning	3
3.2.1	‘plain’ Sectioning	3
3.2.2	‘better’ Sectioning	4
3.3	Useful Macros	4
3.3.1	QUOD ERAT DEMUNSTARNDUM, End of Proof	4
3.3.2	QUOD NON ERAT DEMUNSTARNDUM AT IUCUNDUM EST	4
3.3.3	Stolen Goods	4
3.3.4	Rounding	5
3.4	Grading Table	5
4	Dependencies	6
4.1	Mandatory Dependencies	6
4.2	Recommended Dependencies	7
5	Development and support	8
6	Changelog	8
6.1	Version-Scheme	9

*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

[†]This document corresponds to `homeworkassignment` v2.5, dated 2017/10/15.

7	Examples	10
8	Implementation	10
8.1	Packages & Options	10
8.2	TikZ-Styles	11
8.3	Constants	11
8.4	Geometry	11
8.5	Translations	12
8.6	Headers & Footers	13
9	Redefinition of existing Commands	14
9.1	Internal commands	14
9.1.1	Counter-Commands	14
9.1.2	Counter-Style Parser	15
9.1.3	Counter-Commands II	15
9.1.4	Grading-table	15
9.2	Commands	16

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 customizatuion 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 labled 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>
<code>tikz</code>	Loads TikZ-Package and a couple of Styles, usefull for Papers in Computer-Science and ;athematics. See 8.2 for more informations
<code>fleqn</code>	Passes <code>fleqn</code> to <code>amsmath</code>

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>	is “Kein Kurs” <code>\kurs</code> is deprecated.
<code>\tutorial</code>	Sets the tutorial of the author. Takes it as an argument. Standard Value is
<code>\tutorium</code>	empty, so that this command can be omitted. <code>\tutorium</code> is deprecated.
<code>\deadline</code>	Sets the deadline of the document. Takes it as an argument. Standard value
<code>\abgabe</code>	is <code>\today</code> . <code>\abgabe</code> is deprecated
<code>\sheetTitle</code>	Sets a descriptonal Title of the Sheet, will be written in the header of every page.

3.1.1 Inherited from article

<code>\author</code>	Sets the author of the document.
<code>\date</code>	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

<code>\problem</code>	These commands work like theyr counterpart in article, except that there will be no number, nor will they increase a counter. Nevertheless, hey will be shown in the table of contents.
<code>\subproblem</code>	
<code>\subsubproblem</code>	

`\keyword{#1}` Creates a new Paragraph ,which will start with the Argument in Bold, followed by two non-breaking spaces.
The following Macros make use of `\keyword`, so it is suggested to use them instead.

<code>\solution</code>	They work like <code>\keyword</code> , but take only an optional Argument print out “Solution”, “Proof” “Given”, “To show”, “Assumption”, and “Suppose that”, respectively ¹ , via <code>\keyword</code> . If an argument is passed, they print out this argument after the
<code>\proof</code>	
<code>\given</code>	
<code>\toShow</code>	
<code>\assumption</code>	¹ As of v1.6, Translations are added, depending on the choosen Language, there may be an other Text displayed. See 8.5 for all Translations
<code>\supposeThat</code>	

keyword. They are not mentioned in the table of contents.

3.2.2 ‘better’ Sectioning

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

<code>\newproblem</code>	These commands require no argument, and automatically create a numbered
<code>\newproblem*</code>	title. The optional Argument is the new value for the corresponding counter.
<code>\newsproblem</code>	Normally, <code>\newproblem</code> adds the new Created Problem to the grading-table (see
<code>\newsproblem*</code>	3.4), <code>\newproblem*</code> does not do this.

3.3 Useful Macros

3.3.1 QUOD ERAT DEMUNSTARNDUM, End of Proof

<code>\QED</code>	Display a flushed-right <i>QED</i> , \square , or \blacksquare , respectively. <code>\qed</code> is not implemented, to
<code>\EOP</code>	keep compatibility to several Math-packages, which define the later.
<code>\eop</code>	

3.3.2 QUOD NON ERAT DEMUNSTARNDUM AT IUCUNDUM EST

<code>\QNED</code>	Display a flushed-right \triangle . <code>\QNED</code> displays it in a new line, <code>\qned</code> at the end of
<code>\qned</code>	the same line.
	In Mathematical proofs this symbol is used to mark things, which we did not
	intend to proof, but are interesting anyway.

3.3.3 Stolen Goods

»Das ist alles nur geklaut«

~Tobias Künzel

These Commands are not mine, they are all stolen from Alexander Bartolomey’s²
amath-Class³

<code>\N</code>	Defines a set of mathematical sets, which are verry usefull (see Table 1)		
<code>\Z</code>			
<code>\R</code>	Command	Output	Description
<code>\Q</code>	<code>\N</code>	\mathbb{N}	Natural Numbers
<code>\C</code>	<code>\Z</code>	\mathbb{Z}	Whole Numbers
<code>\F</code>	<code>\Q</code>	\mathbb{Q}	Rational Numbers
<code>\Primes</code>	<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> ⁴	\mathbb{P}	Set of all Primes

Table 1: Field-Commands

²“Occloxiun” on GitHub:<https://github.com/occloxiun>

³amath.sty is part of Alexander Bartolomey’s Alphabet Classes: <https://github.com/occloxiun/AlphabetClasses>

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

Command	Output
<code>\divides and property</code>	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>	$\dot{\cup}$
<code>\diff{<1>}</code>	$\frac{d}{d<1>}$

Table 2: Text-like Functions

`\falls` prints out »falls«⁵

3.3.4 Rounding

Require Mathmode

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

Table 3: Rounding Functions

3.4 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

⁴Has to be `\Primes`, because `\P` is already in use

⁵In German, actual Translation may differ

`\tableofcontent`, it needs a second run of `LATEX`, until all are added.
See example documents for output

4 Dependencies

4.1 Mandatory Dependencies

This class is built upon `article`, so of course the first dependency is:

`article` 1992 LESLIE LAMPORT, 1994-97 FRANK MITTELBACH JOHANNES
BRAAMS, THE `LATEX`-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 `LATEX`-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://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
`LATEX`class and package authors

`array` FRANK MITTELBACH, DAVID CARLISLE, THE `LATEX`-TEAM, [https://
www.ctan.org/pkg/array](https://www.ctan.org/pkg/array),
A new implementations for tables and arrays

`xparse` FRANK MITTELBACH, CHRIS ROWLEY, DAVID CARLISLE, THE `LATEX`3
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 `LATEX 2ε`'s `\newcommand`
macro, with significantly improved functionality.

array possibly can be re-
moved

4.2 Recommended Dependencies

These are not loaded automatically, but require a switch as option (see section 2). 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 subsection 8.2 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

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 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 `\QNED`

v2.0 - 2017/05/23 Change Margins,
Add Option to select older Page-Style,
Change standardlayout to twocolumn and twoside
~~Steal~~ Use Macros by Alexander Bartolomey (See 3.3.3)
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 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

6.1 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.

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_EX2_ε 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 \DeclareBoolOption[false]{tikz}
14 % Redefine the article-options
15 %   \begin{macrocode}
16 \DeclareDefaultOption{\PassOptionsToClass{\CurrentOptionKey}{article}}

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

    Loads required Packages
29 \RequirePackage{suffix}
30 \RequirePackage{fancyhdr}
31 \RequirePackage{xifthen}
32 \RequirePackage{translations}
```

```

33 \PassOptionsToPackage{fleqn}{amsmath}
34 \RequirePackage{amsmath}
35 \RequirePackage{amssymb}
36 \ifhwa@listings
37 \RequirePackage{listings}
38 \lstset{
39   frame = single,
40   breaklines = true,
41   postbreak=\raisebox{0ex}[0ex][0ex]{\ensuremath{\hookrightarrow\space}},
42   basicstyle=\scriptsize
43 }
44 \else
45 \empty
46 \fi
47 \RequirePackage{etoolbox}
48 \RequirePackage{array}
49 \RequirePackage{xparse}
50

```

8.2 TikZ-Styles

If tikz is Wanted, load Usefull Styles

```

51 \ifhwa@tikz
52 \RequirePackage{tikz}
53 \usetikzlibrary{shapes,arrows,positioning,decorations,
54   automata,backgrounds,petri,bending,
55   shapes.multipart}
56 \tikzset{
57   treenode/.style = {shape=circle, rounded corners,
58     draw, align=center},
59   graynode/.style = {fill=gray},
60   normalnode/.style = {treenode, font=\Large, bottom color=white},
61   array/.style = {rectangle split,
62     rectangle split horizontal,
63     rectangle split,
64     draw}
65 }
66 \fi

```

8.3 Constants

Defines some constants

```

67 \newcommand{\hwa@pointboxsize}{3em}

```

8.4 Geometry

Make sure that this is the last Package loaded

```

68 % Make sure that this is the last Package loaded
69 \ifthenelse{equal{\hwa@design}{v2}}{

```

```

70 \RequirePackage{geometry}
71 \ifhwa@twocolumn
72 \geometry{top=2cm, bottom=2cm, left=2cm,
73   headsep=14pt,hmarginratio={1:1}}
74 \else
75 \geometry{top=2cm, bottom=2cm, width=35em,
76   headsep=14pt,hmarginratio={4:3}}
77 \fi
78 }{
79 \ifthenelse{\equal{\hwa@design}{v1}}{
80   \empty
81 }{
82   \ClassError{homeworkassignment}{Value '\hwa@design' for key 'design'
83     is not known}{The option design takes an argument to set the
84     Pagestyle to the one of a previous version. Acceptable values are
85     'v1', or 'v2'}
86 }
87 }

```

8.5 Translations

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

```

88 \DeclareTranslationFallback{aufgabe}{Aufgabe}
89 \DeclareTranslationFallback{loesung}{L"osung}
90 \DeclareTranslationFallback{beweis}{Beweis}
91 \DeclareTranslationFallback{uebungsgruppe}{\ "Ubungsgruppe}
92 \DeclareTranslationFallback{abgabe}{Abgabe}
93 \DeclareTranslationFallback{zuZeigen}{Zu zeigen}
94 \DeclareTranslationFallback{gegeben}{Gegeben}
95 \DeclareTranslationFallback{falls}{falls}
96 \DeclareTranslationFallback{Annahme}{Annahme}
97 \DeclareTranslationFallback{Angenommen-dass}{Angenommen, dass}
98
99 \DeclareTranslation{German}{aufgabe}{Aufgabe}
100 \DeclareTranslation{German}{loesung}{L"osung}
101 \DeclareTranslation{German}{beweis}{Beweis}
102 \DeclareTranslation{German}{uebungsgruppe}{\ "Ubungsgruppe}
103 \DeclareTranslation{German}{abgabe}{Abgabe}
104 \DeclareTranslation{German}{zuZeigen}{Zu zeigen}
105 \DeclareTranslation{German}{gegeben}{Gegeben}
106 \DeclareTranslation{German}{falls}{falls}
107 \DeclareTranslation{German}{Falls}{Falls}
108 \DeclareTranslation{German}{Annahme}{Annahme}
109 \DeclareTranslation{German}{Angenommen-dass}{Angenommen, dass}
110
111 \DeclareTranslation{English}{aufgabe}{Problem}
112 \DeclareTranslation{English}{loesung}{Solution}
113 \DeclareTranslation{English}{beweis}{Proof}
114 \DeclareTranslation{English}{uebungsgruppe}{Tutorial}

```

```

115 \DeclareTranslation{English}{abgabe}{Deadline}
116 \DeclareTranslation{English}{zuZeigen}{To show}
117 \DeclareTranslation{English}{gegeben}{Given}
118 \DeclareTranslation{English}{falls}{if}
119 \DeclareTranslation{English}{Falls}{If}
120 \DeclareTranslation{English}{Annahme}{Assumption}
121 \DeclareTranslation{English}{Angenommen-dass}{Suppose that}

```

8.6 Headers & Footers

Sets the page-headers.

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

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

```

122 \fancypagestyle{firstpage}{
123   %
124   \fancyhf{}
125   % clear all six fields
126   \renewcommand{\headrulewidth}{.7pt}
127   \renewcommand{\footrulewidth}{0pt}
128   \fancyfoot[R]{\thepage}
129   \fancyhead[L]{\hwa@tutorium}
130   \fancyhead[R]{\@date } }
131 \fancypagestyle{followingpage}{
132   \fancyhf{}
133
134   \ifthenelse{\equal{\hwa@design}{v2}}{
135     \ifhwa@twoside % IF
136
137     \fancyhead[R0]{\@author}
138     \fancyhead[L0]{\hwa@kurs\
139       \hwa@tutorium}
140     \fancyhead[LE]{
141       \ifthenelse{\equal{\hwa@sheetTitle}{}}{\hwa@sheetTitle\\}
142       \GetTranslation{abgabe}: \hwa@abgabe
143     }
144     \fancyfoot[R0,LE]{\thepage}
145
146     \else %ELSE
147
148     \fancyhead[R]{\hwa@kurs\
149       \@author}
150     \fancyhead[L]{\hwa@tutorium\
151       \ifthenelse{\equal{\hwa@sheetTitle}{}}{\hwa@sheetTitle\\}
152       \GetTranslation{abgabe}: \hwa@abgabe}
153     \fancyfoot[R]{\thepage}
154     \fi %ENDIF
155   }{
156     % ==== LEGACY CODE; DO NOT CHANGE =====
157     \ifthenelse{\equal{\hwa@design}{v1}}{

```

```

158     \fancyhead[RE,LO]{\@author}
159     \fancyhead[LE,RO]{\hwa@kurs\
160         \GetTranslation{abgabe}: \hwa@abgabe}
161     \fancyfoot[RE,LO]{\thepage}
162 }{
163     \ClassError{homeworkassignment}{Value '\hwa@design' for key 'design'
164         is not known}{The option design takes an argument to set the
165         Pagestyle to the one of a previous version. Acceptable values are
166         'v1', or 'v2'}
167 }
168 % ==== END OF LEGACY CODE =====
169 }
170 \renewcommand{\headrulewidth}{0.7pt}
171 \renewcommand{\footrulewidth}{0pt}
172 }
173 \pagestyle{followingpage}

```

9 Redefinition of existing Commands

D displays equation-numbers as upper-case roman numbers.

```
174 \renewcommand{\theequation}{\Roman{equation}}
```

9.1 Internal commands

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

```

175 \newcommand{\hwa@maketitletext}{
176     \begin{centering}
177         \huge{\textbf{\hwa@kurs}} \vspace{.25cm} {\hrule height 2pt}
178         \vspace{.25cm} \large
179         \ifthenelse{\equal{\hwa@sheetTitle}{}}{\hwa@sheetTitle\\}
180         \GetTranslation{abgabe}: \hwa@abgabe\\
181         \vspace{.5cm} \hrule \vspace{.25cm}
182         \normalsize{\@author}\\
183         \vspace{.25cm} \hrule \vspace{.25cm} \normalsize
184     \end{centering}
185 }

```

9.1.1 Counter-Commands

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

```

186 \newcommand{\hwa@problemno}{\arabic{problem}}
187 \newcommand{\hwa@subproblemno}{\alph{subproblem}}
188 \newcommand{\hwa@subsubproblemno}{\roman{subsubproblem}}

```

9.1.2 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 9.1.3 for example usement.

```

189 \newcommand{\hwa@parseCounterStyle}[3]{
190   \ifthenelse{\equal{#1}{arabic}}{\renewcommand{#2}{\arabic{#3}} }{
191     \ifthenelse{\equal{#1}{roman}}{\renewcommand{#2}{\roman{#3}} }{
192       \ifthenelse{\equal{#1}{alph}}{\renewcommand{#2}{\alph{#3}} }{
193         \ifthenelse{\equal{#1}{Alph}}{\renewcommand{#2}{\Alph{#3}} }{
194           \ifthenelse{\equal{#1}{Roman}}{
195             \renewcommand{#2}{\Roman{#3}} }{
196               \ClassError{homeworkassignment}{Invalid Value #1 for
197                 option Counter-Styling}{Possible Values are alph,
198                 arabic, Arabic, roman or Roman.} } } } } }

```

9.1.3 Counter-Commands II

Counter-Style ParserCommands II Redefines the three counter-commands

```

199 \hwa@parseCounterStyle{\hwa@problemsty}{\hwa@problemno}{problem}
200 \hwa@parseCounterStyle{\hwa@subproblemsty}{\hwa@subproblemno}{subproblem}
201 \hwa@parseCounterStyle{\hwa@subsubproblemsty}{\hwa@subsubproblemno}{subsubproblem}

```

9.1.4 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)

```

202 \edef\hwa@gradingtbl@aux@defs{}
203 \newcommand{\hwa@gradingtbl@aux@lineOne}{}
204 \newcommand{\hwa@gradingtbl@aux@lineTwo}{}
205
206 \edef\hwa@gradingtbl@defs{}
207 \newcommand{\hwa@gradingtbl@lineOne}{}
208 \newcommand{\hwa@gradingtbl@lineTwo}{}

```

\addToGradingTable

```

209 \DeclareDocumentCommand\addToGradingTable{m g}{
210   \edef\hwa@gradingtbl@aux@defs{\hwa@gradingtbl@aux@defs|p{\hwa@pointboxsize}}
211   \edef\hwa@gradingtbl@aux@lineOne{\hwa@gradingtbl@aux@lineOne{#1} &}
212   \IfNoValueTF{#2}{
213     \edef\hwa@gradingtbl@aux@lineTwo{\hwa@gradingtbl@aux@lineTwo &}
214   }{
215     \edef\hwa@gradingtbl@aux@lineTwo{\hwa@gradingtbl@aux@lineTwo\vfill\hfill {#2} &}
216   }
217 }

```

W rite to aux

```

218 \AtEndDocument{%
219   \immediate\write\@auxout{%
220     \gdef\string\hwa@gradingtbl@defs{\hwa@gradingtbl@aux@defs}
221   }
222   \immediate\write\@auxout{%
223     \gdef\string\hwa@gradingtbl@lineOne{\hwa@gradingtbl@aux@lineOne}%
224   }
225   \immediate\write\@auxout{%
226     \gdef\string\hwa@gradingtbl@lineTwo{\hwa@gradingtbl@aux@lineTwo}%
227   }
228 }

```

\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.

```

229 \DeclareDocumentCommand\makeGradingTable{o}{
230   \begin{table}[hb]
231     \centering
232     \large
233     \expandafter\table\expandafter{\hwa@gradingtbl@defs | p{\hwa@pointboxsize} | }\hline
234     \hwa@gradingtbl@lineOne   $\Sigma$          \\ \hline \small
235     \hwa@gradingtbl@lineTwo   \IfNoValueTF{#1}{~}{\vfill\hfill/#1}\vspace{.15cm} \\ \hline
236   \end{table}
237 \end{table}
238 }

```

9.2 Commands

\subject Defines `\kurs`. `\subject` equals `\kurs`

```

239 \newcommand{\hwa@kurs}{?\GetTranslation{subject}??}
240 \newcommand{\subject}[1]{\renewcommand{\hwa@kurs}{#1}}
241 \newcommand{\kurs}[1]{\subject{#1}}

```

\tutorial Defines `\tutorial`. `\tutorial` equals `\tutorial`

```

242 \newcommand{\hwa@tutorial}{?\GetTranslation{uebungsgruppe}??}
243 \newcommand{\tutorial}[1]{\renewcommand{\hwa@tutorial}{#1}}
244 \newcommand{\tutorial}[1]{\tutorial{#1}}

```

\sheetTitle Defines `\sheetTitle`.

```

245 \newcommand{\hwa@sheetTitle}{}
246 \newcommand{\sheetTitle}[1]{\def\hwa@sheetTitle{#1}}

```

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

```

247 \newcommand{\hwa@abgabe}{\today}
248 \newcommand{\deadline}[1]{\def\hwa@abgabe{#1}}
249 \newcommand{\abgabe}[1]{\deadline{#1}}

```


`\maketitle` Overrides maketitle.

```
250
251 \renewcommand{\maketitle} {
252   \thispagestyle{firstpage}
253   \ifhwa@twocolumn{
254     \twocolumn[{
255       \hwa@maketitletext
256     }]
257   }\else{
258     \hwa@maketitletext
259   }\fi
260 }
```

Defines and initialize all counters.

```
261 \newcounter{problem} \setcounter{problem}{0}
262 \newcounter{subproblem}[problem] \setcounter{subproblem}{0}
263 \newcounter{subsubproblem}[subproblem] \setcounter{subsubproblem}{0}
264
```

Defines ‘plain’ sectioning-commands. See 3.2 for more informations.

```
265 \DeclareDocumentCommand\problem{m o}{\@startsection{problem}%Name
266   {1}%Level
267   {\z@}%indent
268   {-2em \@plus -1em \@minus -1em}%beforeskip
269   {1ex \@plus .5ex}%afterskip
270   {\normalfont\Large\bfseries}%style
271   *{#1
272     \IfNoValueF{#2}{
273       \hfill
274       \frame{\framebox[\hwa@pointboxsize]{
275         \hfill \normalfont{\large/\small{#2}}}}
276     }
277   }
278   \addcontentsline{toc}{section}{#1}
279 }
280
281 \DeclareDocumentCommand\subproblem{m o}{\@startsection{subproblem}%Name
282   {2}%Level
283   {\z@}%indent
284   {-1em \@plus -.5em \@minus -.5em}%beforeskip
285   {.5ex \@plus .5ex}%afterskip
286   {\normalfont\large\bfseries}%style
287   *{#1
288     \IfNoValueF{#2}{
289       \hfill \framebox[\hwa@pointboxsize]{
290         \hfill \normalfont\large/\small{#2}}
291     }
292   }
293   \addcontentsline{toc}{subsection}{#1}
294 }
```

```

295
296 \DeclareDocumentCommand\subsubproblem{m o}{\@startsection{subsubproblem}%Name
297   {3}%Level
298   {\z0}%indent
299   {-.5em}%beforeskip
300   {.5em}%afterskip
301   {\normalfont\bfseries}%style
302   *{#1
303     \IfNoValueF{#2}{
304       \hfill \framebox[\hwa@pointboxsize]{
305         \hfill\normalfont\large/\scriptsize{#2}}
306     }
307   }
308 }
309
310 \newcommand{\keyword}[1]{\@startsection{keyword}%Name
311   {4}%Level
312   {\parindent}%indent
313   {-.1em}%beforeskip
314   {\z0}%afterskip
315   {\normalfont\bfseries}%style
316   *{#1~~}
317 }
318
319 \newcommand{\solution}[1][\keyword{\GetTranslation{loesung}\ifstrempy{#1}{\{~#1\}:}}
320
321 \newcommand{\proof}[1][\keyword{\GetTranslation{beweis}\ifstrempy{#1}{\{~#1\}:}}
322
323 \newcommand{\toShow}[1][\keyword{\GetTranslation{zuZeigen}\ifstrempy{#1}{\{~#1\}:}}
324
325 \newcommand{\given}[1][\keyword{\GetTranslation{gegeben}\ifstrempy{#1}{\{~#1\}:}}
326
327 \newcommand{\assumption}[1][\keyword{\GetTranslation{Annahme}\ifstrempy{#1}{\{~#1\}:}}
328
329 \newcommand{\supposeThat}[1][\keyword{\GetTranslation{Angenommen-dass}\ifstrempy{#1}{\{~#1\}:}}
330
331

```

Defines ‘better’ sectioning commands. See 3.2 and 3.2.2 for more informations.

```

332 \DeclareDocumentCommand\newproblem{0{ } g}{
333   \IfNoValueTF{#2}{
334     \newproblem*{#1}
335     \addToGradingTable{\# \hwa@problemno}
336   }{
337     \IfNoValueF{#1}{
338       \setcounter{problem}{1}
339     }
340     %\newproblem*{#1}
341     \problem{\GetTranslation{aufgabe} \hwa@problemno}[#2]
342     \addToGradingTable{\# \hwa@problemno}{/#2}

```

```

343 }
344 }
345
346 \WithSuffix\newcommand\newproblem*[1][\stepcounter{problem}
347 \ifthenelse{equal{#1}{}} { } {\setcounter{problem}{#1}}
348 \problem{\GetTranslation{aufgabe} \hwa@problemno}
349 }
350
351 \DeclareDocumentCommand\newsubproblem{0}{ g}{
352 \stepcounter{subproblem}
353 \ifthenelse{equal{#1}{}} { } {\setcounter{subproblem}{#1}}
354 \IfNoValueTF{#2}{
355 \subproblem{\GetTranslation{aufgabe}
356 \hwa@problemno}{.\hwa@subproblemno}
357 }
358 {
359 \subproblem{\GetTranslation{aufgabe}
360 \hwa@problemno}{.\hwa@subproblemno}[#2]
361 }
362 }
363
364 \DeclareDocumentCommand\newsussubproblem{0}{ g}{
365 \stepcounter{subsubproblem}
366 \ifthenelse{equal{#1}{}} { } {\setcounter{subsubproblem}{#1}}
367 \IfNoValueTF{#2}{
368 \subsubproblem{\hwa@subsubproblemno}}
369 }
370 {
371 \subsubproblem{\hwa@subsubproblemno}[#2]
372 }
373 }
374

```

End of Proof

```

375 \newcommand{\QED}{\begin{flushright}
376 \textsc{Qed}
377 \end{flushright}
378 }
379 \newcommand{\EOP}{\begin{flushright}
380 $\square$
381 \end{flushright}
382 }
383 \newcommand{\eop}{\hfill$\blacksquare$}

```

t demonstrandum at iucundum est

```

384 \newcommand{\QNEDE}{\begin{flushright}
385 $\triangle$
386 \end{flushright}
387 }
388 \newcommand{\qned}{\hfill$\triangle$}

```

Rounding brackets

Round brackets

```

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

```

The following Macros are all stolen (and adapted) from occloxiun (see 3.3.3)

Math Common Set Symbols

```

393 \newcommand{\N}{\ensuremath{\mathbb{N}}}
394 \newcommand{\Z}{\ensuremath{\mathbb{Z}}}
395 \newcommand{\R}{\ensuremath{\mathbb{R}}}
396 \newcommand{\Q}{\ensuremath{\mathbb{Q}}}
397 \newcommand{\C}{\ensuremath{\mathbb{C}}}
398 \newcommand{\F}{\ensuremath{\mathbb{F}}}
399 % The last one is mine
400 \newcommand{\Primes}{\ensuremath{\mathbb{P}}}

```

Mathematical Functions

```

401 \DeclareMathOperator{\GL}{GL}
402 \DeclareMathOperator{\id}{id}
403 \DeclareMathOperator{\Var}{Var}
404 \DeclareMathOperator{\Perm}{Perm}
405 \DeclareMathOperator{\MComb}{MComb}
406 \DeclareMathOperator{\Comb}{Comb}
407 \DeclareMathOperator{\Pot}{Pot}
408 \DeclareMathOperator{\Map}{Map}
409 \DeclareMathOperator{\Hom}{Hom}
410 \DeclareMathOperator{\Ker}{Ker}
411 \DeclareMathOperator{\Intpol}{Intpol}
412 \DeclareMathOperator{\Pol}{Pol}
413 \DeclareMathOperator{\Sol}{Sol}
414 \DeclareMathOperator{\Bin}{Bin}
415 \DeclareMathOperator{\charakteristik}{char}
416 \newcommand{\diff}[1]{\ensuremath{\frac{d}{d\#1}}}
417 \newcommand{\dx}{\mathrm{d}x}
418
419 \newcommand{\divides}{\ensuremath{\mid}}
420 \newcommand{\property}{\ensuremath{\parallel}}
421
422 \renewcommand{\dim}[1][\mathrm{dim}]{\text{\dim}_{#1}}
423 \renewcommand{\Im}{\text{\Im}}
424
425 \newcommand{\excup}{\ensuremath{\stackrel{\cdot}{\cup}}}
426
427 \newcommand{\falls}{\text{\ \GetTranslation{falls}}}

```

Math Big Quantors

```

428 \let\oforall\forall
429 \let\oexists\exists
430 \renewcommand{\forall}{\ensuremath{\hspace{2pt}\forall\hspace{2pt}}}
431 \renewcommand{\exists}{\ensuremath{\hspace{2pt}\exists\hspace{2pt}}}
432 \newcommand{\bigforall}{\mbox{\raisebox{-2pt}[\height][\depth]{\Large $\mathsurround{4pt}\forall$}}}
433 \newcommand{\bigexists}{\mbox{\raisebox{-2pt}[\height][\depth]{\Large $\mathsurround{4pt}\exists$}}}

The End
434 \endinput

```