The homeworkassignment[*]class[†]

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

November 7, 2017

Contents

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

[†] This document corresponds to homeworkassignment v2.5b, dated 2017/11/07.

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 verry best.

2 Options

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

`design=<1>` Allows the User to select an older page-style, for backwards compatibility.
Recognized values are `v1` and `v2`. 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, *older styles are not going to be maintained!*

`tikz` Loads TikZ-Package and a couple of Styles, usefull for Papers in Computer-Science and ;athematics. See 8.2 for more informations

`fleqn` Passes `fleqn` to amsmath

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

`\subject`
`\kurs`

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

`\tutorial`
`\tutorium`

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

`\deadline`
`\abgabe`

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

`\sheetTitle`

Sets a descriptional Title of the Sheet, will be written in the header of every page.

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`
`\subproblem`
`\subsubproblem`

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.

`\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.

`\solution`
`\proof`
`\given`
`\toShow`
`\assumption`
`\supposeThat`

They work like `\keyword`, but take only an optional Argument print out “Solution”, “Proof” “Given”, “To show”, “Assumption”, and “Suppose that”, respectively [1], via `\keyword`. If an argument is passed, they print out this argument after the

[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

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.

`\newproblem`
`\newproblem*`
`\newsubproblem`
`\newsubsubproblem`

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

3.3 Useful Macros

3.3.1 Quod Erat Demunstarndum, End of Proof

`\QED`
`\EOP`
`\eop`

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

3.3.2 Quod Non Erat Demunstarndum at iucundum est

`\QNED`
`\qned`

Display a flushed-right △. `\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.

3.3.3 Stolen Goods

»Das ist alles nur geklaut«

~Tobias Künzel

These Commands are not mine, they are all stolen from Alexander Bartolomey's[2] `amath`-Class[3]

`\N`
`\Z`
`\R`
`\Q`
`\C`
`\F`
`\Primes`

Defines a set of mathematical sets, which are verry usefull (see Table 1)

Command	Output	Description
`\N`	$\mathbb{N}$	Natural Numbers
`\Z`	$\mathbb{Z}$	Whole Numbers
`\Q`	$\mathbb{Q}$	Rational Numbers
`\R`	$\mathbb{R}$	Real Numbers
`\C`	$\mathbb{C}$	Complex Numbers
`\F_n`	$\mathbb{F}_n$	Prime Field to base n
`\Primes`[4]	$\mathbb{P}$	Set of all Primes

Table 1: Field-Commands

[2] "Occloxium" on GitHub:`https://github.com/occloxium`

[3] `amath.sty` is part of Alexander Bartolomey's Alphabet Classes: `https://github.com/occloxium/AlphabetClasses`

`Functions and Operators` Output usefull Plaintext-Operators and Functions. See table 2. Require Mathmode

Command	Output
`\divides` and `property`	Prints a vertical Bar
`\Var`	Var
`\Perm`	Perm
`\Comb`	Comb
`\MComb`	MComb
`\Pot`	Pot
`\Map`	Map
`\Bin`	Bin
`\GL`	GL
`\id`	id
`\dx`	dx
`\excup`	$\dot{\cup}$
`\diff{<1>}`	$\frac{d}{d<1>}$

Table 2: Text-like Functions

`\falls` prints out »falls«[5]

3.3.4 Rounding

Require Mathmode

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

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?

`\addToGradingTable` Adds the given parameter as an excercise to the Grading-Table. All Problems, created with `\newproblem` are added automatically.

`\makeGradingTable` Prints out the Table containig all Defined exercises (≠Problems). Like

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

[5] In German, actual Translation may differ

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

4 Dependencies

4.1 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 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`,
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 LaTeXclass and package authors

`array` Frank Mittelbach, David Carlisle, The LaTeX–Team, `https://www.ctan.org/pkg/array`,
A new implementations for tables and arrays

`array` possibly can be removed

`xparse` Frank Mittelbach, Chris Rowley, David Carlisle, The LaTeX3 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 2_ε’s `\newcommand` macro, with significantly improved functionality.

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

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

`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

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 developtment 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 developtment, developtment-versions do not have numeric patch–numbers, but alphabetic identifiers, directly after the minor–version.

7 Examples

For examples, pleas see the Git-Repo at `https://github.com/ACHinrichs/LaTeX-templates`

8 Implementation

The following part is verry boring, but I have not found a solution to create a `.cls`–file without including the implemetation into the document. Loads LaTeX2e and sets the Version Loads the `article`, which is the base-class.

8.1 Packages & Options

```
\RequirePackage{kvoptions}
\SetupKeyvalOptions{ family=hwa,
  prefix=hwa@ }
\DeclareStringOption[arabic]{problemsty}
\DeclareStringOption[alph]{subproblemsty}
\DeclareStringOption[roman]{subsubproblemsty}
\DeclareBoolOption[false]{listings}
\DeclareStringOption[v2]{design}
\DeclareBoolOption[true]{twoside}
\DeclareComplementaryOption{oneside}{twoside}
\DeclareBoolOption[true]{twocolumn}
\DeclareComplementaryOption{onecolumn}{twocolumn}
\DeclareBoolOption[false]{tikz}
\DeclareStringOption[all]{hlines}
% Redefine the article-options
%    \begin{macrocode}
\DeclareDefaultOption{\PassOptionsToClass{\CurrentOptionKey}{article}}
```

Processes the Options and loades article

```
\ProcessKeyvalOptions*
\ifhwa@twoside
\PassOptionsToClass{twoside}{article}
\else
\PassOptionsToClass{oneside}{article}
\fi
\ifhwa@twocolumn
\PassOptionsToClass{twocolumn}{article}
\else
\PassOptionsToClass{onecolumn}{article}
\fi
\LoadClass{article}
```

Loads required Packages

```
\RequirePackage{suffix}
\RequirePackage{fancyhdr}
\RequirePackage{xifthen}
```

```
\RequirePackage{translations}
\PassOptionsToPackage{fleqn}{amsmath}
\RequirePackage{amsmath}
\RequirePackage{amssymb}
\ifhwa@listings
\RequirePackage{listings}
\lstset{
  frame = single,
  breaklines = true,
  postbreak=\raisebox{0ex}[0ex][0ex]{\ensuremath{\hookrightarrow\space}},
  basicstyle=\scriptsize
}
\else
\empty
\fi
\RequirePackage{etoolbox}
\RequirePackage{array}
\RequirePackage{xparse}

```

\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

```

\newcommand{\hwa@hline@LONE}{\vspace{.25cm} {\hrule height 2pt}
  \vspace{.25cm}}
\newcommand{\hwa@hline@LTWO}{\vspace{.5cm} \hrule \vspace{.25cm}}
\newcommand{\hwa@headrulewidth}{.7pt}
\ifthenelse{\equal{\hwa@hlines}{all}}{
  \renewcommand{\hwa@hline@LONE}{\vspace{.25cm} {\hrule height 2pt}
    \vspace{.25cm}}
  \renewcommand{\hwa@headrulewidth}{.7pt}
  \renewcommand{\hwa@hline@LTWO}{\vspace{.5cm} \hrule \vspace{.25cm}}
}{
  \ifthenelse{\equal{\hwa@hlines}{decreased}}{
    \renewcommand{\hwa@hline@LONE}{ \vspace{.25cm} {\hrule height 2pt}
      \vspace{.25cm}}
    \renewcommand{\hwa@headrulewidth}{.7pt}
  }{\ifthenelse{\equal{\hwa@hlines}{header}}{
      \renewcommand{\hwa@headrulewidth}{.7pt}
    }{\ifthenelse{\equal{\hwa@hlines}{none}}{
        \renewcommand{\hwa@headrulewidth}{0pt}
      }{
        \ClassError{homeworkassignment}{Value ’\hwa@lines’ for key ’hlines’
          is not known}{The option hlines takes an argument to set which
          hlines are drawn. Possible values are ’all’,’decreased’ , ’header’, and
          'none'. 'all' is standard.}
      }
    }
    \renewcommand{\hwa@hline@LONE}{~\\\vspace{.5cm}}
  }
  \renewcommand{\hwa@hline@LTWO}{\vspace{.75cm}}
}
```

8.2 TikZ-Styles

If tikz is Wanted, load Usefull Styles

```
\ifhwa@tikz
\RequirePackage{tikz}
\usetikzlibrary{shapes,arrows,positioning,decorations,
  automata,backgrounds,petri,bending,
  shapes.multipart}
\tikzset{
  treenode/.style = {shape=circle, rounded corners,
    draw, align=center},
  graynode/.style = {fill=gray},
  normalnode/.style     = {treenode, font=\Large, bottom color=white},
  array/.style = {rectangle split,
    rectangle split horizontal,
    rectangle split,
    draw}
}
\fi
```

8.3 Constants

Defines some constants

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

8.4 Geometry

Make sure that this is the last Package loaded

```
 % Make sure that this is the last Package loaded
\ifthenelse{\equal{\hwa@design}{v2}}{
  \RequirePackage{geometry}
  \ifhwa@twocolumn
  \geometry{top=2cm, bottom=2cm, left=2cm,
    headsep=14pt,hmarginratio={1:1}}
  \else
  \geometry{top=2cm, bottom=2cm, width=35em,
    headsep=14pt,hmarginratio={4:3}}
  \fi
}{
  \ifthenelse{\equal{\hwa@design}{v1}}{
    \empty
  }{
    \ClassError{homeworkassignment}{Value ’\hwa@design’ for key ’design’
    is not known}{The option design takes an argument to set the
    Pagestyle to the one of a previous version. Acceptable values are
    ’v1’, or ’v2’}
  }
}
```

8.5 Translations

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

```
\DeclareTranslationFallback{aufgabe}{Aufgabe}
\DeclareTranslationFallback{loesung}{L\"osung}
\DeclareTranslationFallback{beweis}{Beweis}
\DeclareTranslationFallback{uebungsgruppe}{\"Ubungsgruppe}
\DeclareTranslationFallback{abgabe}{Abgabe}
\DeclareTranslationFallback{zuZeigen}{Zu zeigen}
\DeclareTranslationFallback{gegeben}{Gegeben}
\DeclareTranslationFallback{falls}{falls}
\DeclareTranslationFallback{Annahme}{Annahme}
\DeclareTranslationFallback{Angenommen-dass}{Anngenommen, dass}

\DeclareTranslation{German}{aufgabe}{Aufgabe}
\DeclareTranslation{German}{loesung}{L\"osung}
\DeclareTranslation{German}{beweis}{Beweis}
\DeclareTranslation{German}{uebungsgruppe}{\"Ubungsgruppe}
\DeclareTranslation{German}{abgabe}{Abgabe}
\DeclareTranslation{German}{zuZeigen}{Zu zeigen}
\DeclareTranslation{German}{gegeben}{Gegeben}
\DeclareTranslation{German}{falls}{falls}
\DeclareTranslation{German}{Falls}{Falls}
\DeclareTranslation{German}{Annahme}{Annahme}
\DeclareTranslation{German}{Angenommen-dass}{Anngenommen, dass}

\DeclareTranslation{English}{aufgabe}{Problem}
\DeclareTranslation{English}{loesung}{Solution}
\DeclareTranslation{English}{beweis}{Proof}
\DeclareTranslation{English}{uebungsgruppe}{Tutorial}
\DeclareTranslation{English}{abgabe}{Deadline}
\DeclareTranslation{English}{zuZeigen}{To show}
\DeclareTranslation{English}{gegeben}{Given}
\DeclareTranslation{English}{falls}{if}
\DeclareTranslation{English}{Falls}{If}
\DeclareTranslation{English}{Annahme}{Assumption}
\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.

```
\fancypagestyle{firstpage}{
  %
  \fancyhf{}
  % clear all six fields
  \renewcommand{\headrulewidth}{\hwa@headrulewidth}
  \renewcommand{\footrulewidth}{0pt}
  \fancyfoot[R]{\thepage}
  \fancyhead[L]{\hwa@tutorium}
  \fancyhead[R]{\@date } }
\fancypagestyle{followingpage}{
  \fancyhf{}

  \ifthenelse{\equal{\hwa@design}{v2}}{
    \ifhwa@twoside % IF

    \fancyhead[RO]{\@author}
    \fancyhead[LO]{\hwa@kurs\\
      \hwa@tutorium}
    \fancyhead[LE]{
      \ifthenelse{\equal{\hwa@sheetTitle}{}}{}{\hwa@sheetTitle\\}
      \GetTranslation{abgabe}: \hwa@abgabe
    }
    \fancyfoot[RO,LE]{\thepage}

    \else %ELSE

    \fancyhead[R]{\hwa@kurs\\
      \@author}
    \fancyhead[L]{\hwa@tutorium\\
      \ifthenelse{\equal{\hwa@sheetTitle}{}}{}{\hwa@sheetTitle\\}
      \GetTranslation{abgabe}: \hwa@abgabe}
    \fancyfoot[R]{\thepage}
    \fi %ENDIF
  }{
    % ==== LEGACY CODE; DO NOT CHANGE ================================
    \ifthenelse{\equal{\hwa@design}{v1}}{
      \fancyhead[RE,LO]{\@author}
      \fancyhead[LE,RO]{\hwa@kurs\\
        \GetTranslation{abgabe}: \hwa@abgabe}
      \fancyfoot[RE,LO]{\thepage}
    }{
      \ClassError{homeworkassignment}{Value '\hwa@design' for key 'design'
        is not known}{The option design takes an argument to set the
        Pagestyle to the one of a previous version. Acceptable values are
        'v1', or 'v2'}
    }
    % ==== END OF LEGACY CODE ========================================
  }
  \renewcommand{\headrulewidth}{\hwa@headrulewidth}
  \renewcommand{\footrulewidth}{0pt}
}
\pagestyle{followingpage}
```

9 Redefinition of existing Commands

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

```
\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

```
\newcommand{\hwa@maketitletext}{
  \begin{centering}
    \huge{\textbf{\hwa@kurs}}\hwa@hline@LONE \large
    \ifthenelse{\equal{\hwa@sheetTitle}{}}{}{\hwa@sheetTitle\\}
    \GetTranslation{abgabe}: \hwa@abgabe\\
    \hwa@hline@LTWO
    \normalsize{\@author}\\
    \hwa@hline@LTWO \normalsize
  \end{centering}
}
```

9.1.1 Counter–Commands

Counter-Commands

These are used to output the Exercise numbers in the desired style

```
\newcommand{\hwa@problemno}{\arabic{problem}}
\newcommand{\hwa@subproblemno}{\alph{subproblem}}
\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 coresponding counter (#3) to redefine #1, so that it corresponds to #2. See 9.1.3 for example usement.

```
\newcommand{\hwa@parseCounterStyle}[3]{
  \ifthenelse{\equal{#1}{arabic}}{ \renewcommand{#2}{\arabic{#3}} }{
    \ifthenelse{\equal{#1}{roman}}{ \renewcommand{#2}{\roman{#3}} }{
      \ifthenelse{\equal{#1}{alph}}{ \renewcommand{#2}{\alph{#3}} }{
        \ifthenelse{\equal{#1}{Alph}}{ \renewcommand{#2}{\Alph{#3}} }{
          \ifthenelse{\equal{#1}{Roman}}{
            \renewcommand{#2}{\Roman{#3}} }{
```

```
            \ClassError{homeworkassignment}{Invalid Value #1 for
              option Counter-Styling}{Possible Values are alph,
              arabic, Arabic, roman or Roman.}  } } } } } }
```

9.1.3 Counter–Commands II

Counter-Style ParserCommands II

Redefines the three counter-commands

```
\hwa@parseCounterStyle{\hwa@problemsty}{\hwa@problemno}{problem}
\hwa@parseCounterStyle{\hwa@subproblemsty}{\hwa@subproblemno}{subproblem}
\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)

```
\edef\hwa@gradingtbl@aux@defs{}
\newcommand{\hwa@gradingtbl@aux@lineOne}{}
\newcommand{\hwa@gradingtbl@aux@lineTwo}{}

\edef\hwa@gradingtbl@defs{}
\newcommand{\hwa@gradingtbl@lineOne}{}
\newcommand{\hwa@gradingtbl@lineTwo}{}
```

\addToGradingTable

```
\DeclareDocumentCommand\addToGradingTable{m g}{
  \edef\hwa@gradingtbl@aux@defs{\hwa@gradingtbl@aux@defs|p{\hwa@pointboxsize}}
  \edef\hwa@gradingtbl@aux@lineOne{\hwa@gradingtbl@aux@lineOne{#1} &}
  \IfNoValueTF{#2}{
    \edef\hwa@gradingtbl@aux@lineTwo{\hwa@gradingtbl@aux@lineTwo &}
  }{
    \edef\hwa@gradingtbl@aux@lineTwo{\hwa@gradingtbl@aux@lineTwo\vfill\hfill
      {\string\small #2} &}
  }
}
```

W rite to aux

```
\AtEndDocument{%
  \immediate\write\@auxout{%
    \gdef\string\hwa@gradingtbl@defs{\hwa@gradingtbl@aux@defs}
  }
  \immediate\write\@auxout{%
    \gdef\string\hwa@gradingtbl@lineOne{\hwa@gradingtbl@aux@lineOne}%
  }
  \immediate\write\@auxout{%
    \gdef\string\hwa@gradingtbl@lineTwo{\hwa@gradingtbl@aux@lineTwo}%
  }
}
```

`\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.
Displayes the total number of maximum Problems, if given by Argument

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

```
\DeclareDocumentCommand\makeGradingTable{o}{
  \begin{table}[hb]
    \centering
    \large
    \expandafter\tabular\expandafter{\hwa@gradingtbl@defs ||p{\hwa@pointboxsize}|}\hline
    \hwa@gradingtbl@lineOne  $\Sigma$       \\\hline\small
    \hwa@gradingtbl@lineTwo  \IfNoValueTF{#1}{~}{\vfill\hfill/#1}\vspace{.15cm}\\\hline
    \endtabular
  \end{table}
  }
```

9.2 Commands

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

```
\newcommand{\hwa@kurs}{?\GetTranslation{subject}?}
\newcommand{\subject}[1]{\renewcommand{\hwa@kurs}{#1}}
\newcommand{\kurs}[1]{\subject{#1}}
```

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

```
\newcommand{\hwa@tutorium}{?\GetTranslation{uebungsgruppe}?}
\newcommand{\tutorial}[1]{\renewcommand{\hwa@tutorium}{#1}}
\newcommand{\tutorium}[1]{\tutorial{#1}}
```

`\sheetTitle` Defines `\sheetTitle`.

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

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

```
\newcommand{\hwa@abgabe}{\today}
\newcommand{\deadline}[1]{\def\hwa@abgabe{#1}}
\newcommand{\abgabe}[1]{\deadline{#1}}
```

`\maketitle` Overrides maketitle.

```

\renewcommand{\maketitle} {
  \thispagestyle{firstpage}
  \ifhwa@twocolumn{
    \twocolumn[{
      \hwa@maketitletext
    }]
  }\else{
    \hwa@maketitletext
  }\fi
}
```

Defines and initialize all counters.

```
\newcounter{problem} \setcounter{problem}{0}
\newcounter{subproblem}[problem] \setcounter{subproblem}{0}
\newcounter{subsubproblem}[subproblem] \setcounter{subsubproblem}{0}

```

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

```
\DeclareDocumentCommand\problem{m o}{\@startsection{problem}%Name
  {1}%Level
  {\z@}%indent
  {-2em \@plus -1em \@minus -1em}%beforeskip
  {1ex \@plus .5ex}%afterskip
  {\normalfont\Large\bfseries}%style
  *{#1
    \IfNoValueF{#2}{
      \hfill
     \frame{\framebox[\hwa@pointboxsize]{
         \hfill \normalfont{\large/\small{#2}}}}
    }
  }
  \addcontentsline{toc}{section}{#1}
}

\DeclareDocumentCommand\subproblem{m o}{\@startsection{subproblem}%Name
  {2}%Level
  {\z@}%indent
  {-1em \@plus -.5em \@minus -.5em}%beforeskip
  {.5ex \@plus .5ex}%afterskip
  {\normalfont\large\bfseries}%style
  *{#1
    \IfNoValueF{#2}{
      \hfill \framebox[\hwa@pointboxsize]{
        \hfill\normalfont\large/\small{#2}}
    }
  }
  \addcontentsline{toc}{subsection}{#1}
}

\DeclareDocumentCommand\subsubproblem{m o}{\@startsection{subsubproblem}%Name
  {3}%Level
  {\z@}%indent
  {-.5em}%beforeskip
  {.5em}%afterskip
  {\normalfont\bfseries}%style
  *{#1
    \IfNoValueF{#2}{
      \hfill \framebox[\hwa@pointboxsize]{
        \hfill\normalfont\large/\scriptsize{#2}}
    }
  }
}

\newcommand{\keyword}[1]{\@startsection{keyword}%Name
  {4}%Level
  {\parindent}%indent
  {-.1em}%beforeskip
  {\z@}%afterskip
  {\normalfont\bfseries}%style
  *{#1~~}
}

\newcommand{\solution}[1][]{\keyword{\GetTranslation{loesung}\ifstrempty{#1}{}{~#1}:}}

\newcommand{\proof}[1][]{\keyword{\GetTranslation{beweis}\ifstrempty{#1}{}{~#1}:}}

\newcommand{\toShow}[1][]{\keyword{\GetTranslation{zuZeigen}\ifstrempty{#1}{}{~#1}:}}

\newcommand{\given}[1][]{\keyword{\GetTranslation{gegeben}\ifstrempty{#1}{}{~#1}:}}

\newcommand{\assumption}[1][]{\keyword{\GetTranslation{Annahme}\ifstrempty{#1}{}{~#1}:}}

\newcommand{\supposeThat}[1][]{\keyword{\GetTranslation{Angenommen-dass}\ifstrempty{#1}{}{~#1}}


```

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

```
\DeclareDocumentCommand\newproblem{O{} g}{
  \IfNoValueTF{#2}{
    \newproblem*[#1]
    \addToGradingTable{\# \hwa@problemno}
  }{
    \IfNoValueF{#1}{
      \setcounter{problem}{#1}
    }
    %\newproblem*[#1]
    \problem{\GetTranslation{aufgabe} \hwa@problemno}[#2]
    \addToGradingTable{\# \hwa@problemno}{/#2}
  }
}

\WithSuffix\newcommand\newproblem*[1][]{\stepcounter{problem}
  \ifthenelse{\equal{#1}{}} { } {\setcounter{problem}{#1}}
  \problem{\GetTranslation{aufgabe} \hwa@problemno}
}

\DeclareDocumentCommand\newsubproblem{O{} g}{
  \stepcounter{subproblem}
  \ifthenelse{\equal{#1}{}} { } {\setcounter{subproblem}{#1}}
  \IfNoValueTF{#2}{
    \subproblem{\GetTranslation{aufgabe}
      \hwa@problemno{}.\hwa@subproblemno}
  }
  {
    \subproblem{\GetTranslation{aufgabe}
      \hwa@problemno{}.\hwa@subproblemno}[#2]
  }
}

\DeclareDocumentCommand\newsubsubproblem{O{} g}{
  \stepcounter{subsubproblem}
  \ifthenelse{\equal{#1}{}} { } {\setcounter{subsubproblem}{#1}}
  \IfNoValueTF{#2}{
    \subsubproblem{\hwa@subsubproblemno)}
  }
  {
    \subsubproblem{\hwa@subsubproblemno)}[#2]
  }
}

```

End of Proof

```
\newcommand{\QED}{\begin{flushright}
    \textsc{Qed}
  \end{flushright}
}
\newcommand{\EOP}{\begin{flushright}
    $\square$
  \end{flushright}
}
\newcommand{\eop}{\hfill$\blacksquare$}
```

t demonstrandum at iucundum est

```
\newcommand{\QNED}{\begin{flushright}
    $\triangle$
  \end{flushright}
}
\newcommand{\qned}{\hfill$\triangle$}
```

Rounding brakets

Round brakets

```
\newcommand{\floor}[1]{\ensuremath{\left\lfloor #1 \right\rfloor}}
\newcommand{\ceil}[1]{\ensuremath{\left\lceil #1 \right\rceil}}
\newcommand{\roundHU}[1]{\ensuremath{\left\lceil #1 \right\rfloor}}
\newcommand{\roundHD}[1]{\ensuremath{\left\lfloor #1 \right\rceil}}
```

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

Math Common Set Symbols

```
\newcommand{\N}{\ensuremath{\mathbb{N}}}
\newcommand{\Z}{\ensuremath{\mathbb{Z}}}
\newcommand{\R}{\ensuremath{\mathbb{R}}}
\newcommand{\Q}{\ensuremath{\mathbb{Q}}}
\newcommand{\C}{\ensuremath{\mathbb{C}}}
\newcommand{\F}{\ensuremath{\mathbb{F}}}
% The last one is mine
\newcommand{\Primes}{\ensuremath{\mathbb{P}}}
```

Mathematical Functions

```
\DeclareMathOperator{\GL}{GL}
\DeclareMathOperator{\id}{id}
\DeclareMathOperator{\Var}{Var}
\DeclareMathOperator{\Perm}{Perm}
\DeclareMathOperator{\MComb}{MComb}
\DeclareMathOperator{\Comb}{Comb}
\DeclareMathOperator{\Pot}{Pot}
\DeclareMathOperator{\Map}{Map}
\DeclareMathOperator{\Hom}{Hom}
\DeclareMathOperator{\Ker}{Ker}
\DeclareMathOperator{\Intpol}{Intpol}
\DeclareMathOperator{\Pol}{Pol}
\DeclareMathOperator{\Sol}{Sol}
\DeclareMathOperator{\Bin}{Bin}
\DeclareMathOperator{\charakteristik}{char}
\newcommand{\diff}[1]{\ensuremath{\frac{d}{d#1}}}
\newcommand{\dx}{\:dx}

\newcommand{\divides}{\ensuremath{\ |\ }}
\newcommand{\property}{\ensuremath{\ |\ }}

\renewcommand{\dim}[1][]{\ensuremath{\text{dim}_{#1}\ }}
\renewcommand{\Im}{\ensuremath{\text{Im}\ }}

\newcommand{\excup}{\ensuremath{\stackrel{.}{\cup}}}

\newcommand{\falls}{\text{\ \GetTranslation{falls}}\ }
```

Math Big Quantors

```
\let\oforall\forall
\let\oexists\exists
\renewcommand{\forall}{\ensuremath{\hskip 2pt \oforall \hskip 2pt}}
\renewcommand{\exists}{\ensuremath{\hskip 2pt \oexists \hskip 2pt}}
\newcommand{\bigforall}{\mbox{\raisebox{-2pt}[\height][\depth]{\Large $\mathsurround4pt\forall$
\newcommand{\bigexists}{\mbox{\raisebox{-2pt}[\height][\depth]{\Large $\mathsurround4pt\exists$
```

The End

```
\endinput
```