`hwexam.sty/cls`: An Infrastructure for formatting Assignments and Exams*

Michael Kohlhase
FAU Erlangen-Nürnberg
http://kwarc.info/kohlhase

March 20, 2019

Abstract

The `hwexam` package and class allows individual course assignment sheets and compound assignment documents using problem files marked up with the `problem` package.

Contents

---

*Version v1.1 (last revised 2019/03/20)

1 Introduction

The `hwexam` package and class supplies an infrastructure that allows to format nice-looking assignment sheets by simply including problems from problem files marked up with the `problem` package [Koh16c]. It is designed to be compatible with `problems.sty`, and inherits some of the functionality.

2 The User Interface

2.1 Package and Class Options

mh The `hwexam` class takes the `mh` option that turns on MathHub support.

The `hwexam` package and class take the options `solutions`, `notes`, `hints`, `pts`, `min`, and `boxed` that are just passed on to the `problems` package (cf. its documentation for a description of the intended behavior).

showmeta If the `showmeta` option is set, then the metadata keys are shown (see [Koh16a] for details and customization options).

The `hwexam` class additionally accepts the options `report`, `book`, `chapter`, `part`, and `showignores`, of the `omdoc` package [Koh16b] on which it is based and passes them on to that. For the `extrefs` option see [Koh16d].

2.2 Assignments

assignment This package supplies the `assignment` environment that groups problems into
number assignment sheets. It takes an optional KeyVal argument with the keys `number` (for the assignment number; if none is given, 1 is assumed as the default or — in multi-assignment documents — the ordinal of the `assignment` environment),
title `title` (for the assignment title; this is referenced in the title of the assignment
type sheet), `type` (for the assignment type; e.g. "quiz", or "homework"), `given` (for
given the date the assignment was given), and `due` (for the date the assignment is due).
due

2.3 Typesetting Exams

multiple Furthermore, the `hwexam` package takes the option `multiple` that allows to combine multiple assignment sheets into a compound document (the assignment sheets are treated as section, there is a table of contents, etc.).

test Finally, there is the option `test` that modifies the behavior to facilitate formatting tests. Only in `test` mode, the macros `\testspace`, `\testnewpage`, and `\testemptypage` have an effect: they generate space for the students to solve the given problems. Thus they can be left in the LaTeX source.

\testspace `\testspace` takes an argument that expands to a dimension, and leaves ver-
\testnewpage tical space accordingly. `\testnewpage` makes a new page in `test` mode, and
\testemptypage `\testemptypage` generates an empty page with the cautionary message that this page was intentionally left empty.

testheading Finally, the `\testheading` takes an optional keyword argument where the keys
duration `duration` specifies a string that specifies the duration of the test, `min` specifies the
min

equivalent in number of minutes, and `reqpts` the points that are required for a perfect grade.

```
\title{320101 General Computer Science (Fall 2010)}
\begin{testheading}[duration=one hour,min=60,reqpts=27]
  Good luck to all students!
\end{testheading}
```

formats to

Name: Matriculation Number:

320101 General Computer Science (Fall 2010)

March 20, 2019

You have one hour(sharp) for the test;
Write the solutions to the sheet.
The estimated time for solving this exam is 58 minutes, leaving you 2 minutes for revising your exam.
You can reach 30 points if you solve all problems. You will only need 27 points for a perfect score, i.e. 3 points are bonus points.

Different problems test different skills and knowledge, so do not get stuck on one problem.

prob.	1.1	2.1	2.2	2.3	3.1	3.2	3.3	Sum	grade
total	4	4	6	6	4	4	2	30	
reached									

To be used for grading, do not write here

good luck

Example 1: A generated test heading.

2.4 Including Assignments

The `\includeassignment` macro can be used to include an assignment from another file. It takes an optional KeyVal argument and a second argument which is a path to the file containing the problem (the macro assumes that there is only one `assignment` environment in the included file). The keys `number`, `title`, `type`, `given`, and `due` are just as for the `assignment` environment and (if given) overwrite the ones specified in the `assignment` environment in the included file.

3   Limitations

In this section we document known limitations. If you want to help alleviate them, please feel free to contact the package author. Some of them are currently discussed in the sTeX GitHub repository [sTeX].

1. none reported yet.

4 Implementation: The hwexam Class

The functionality is spread over the `hwexam` class and package. The class provides the `document` environment and pre-loads some convenience packages, whereas the package provides the concrete functionality.

4.1 Class Options

To initialize the `hwexam` class, we declare and process the necessary options by passing them to the respective packages and classes they come from.

```
1 ⟨*cls⟩
2 \DeclareOption*{
3   \PassOptionsToClass{\CurrentOption}{omdoc}
4   \PassOptionsToPackage{\CurrentOption}{stex}
5   \PassOptionsToPackage{\CurrentOption}{hwexam}
6   \PassOptionsToPackage{\CurrentOption}{tikzinput}
7 }
8 \ProcessOptions
```

We load `omdoc.cls`, and the desired packages. For the LaTeXML bindings, we make sure the right packages are loaded.

```
9  \LoadClass{omdoc}
10 \RequirePackage{stex}
11 \RequirePackage{hwexam}
12 \RequirePackage{tikzinput}
13 \RequirePackage{graphicx}
14 \RequirePackage{a4wide}
15 \RequirePackage{amssymb}
16 \RequirePackage{amstext}
17 \RequirePackage{amsmath}
```

Finally, we register another keyword for the `document` environment. We give a default assignment type to prevent errors

```
18 \newcommand\assig@default@type{\hwexam@assignment@kw}
19 \addmetakey[\assig@default@type]{document}{hwexamtype}
20 \def\document@hwexamtype{\assig@default@type}
21 ⟨/cls⟩
```

5 Implementation: The hwexam Package

5.1 Package Options

The first step is to declare (a few) package options that handle whether certain information is printed or not. Some come with their own conditionals that are set by the options, the rest is just passed on to the `problems` package.

```
22 ⟨*package⟩
23 \newif\if@hwexam@mh@\@hwexam@mh@false
```

```
24 \DeclareOption{mh}{\@hwexam@mh@true}
25 \newif\iftest\testfalse
26 \DeclareOption{test}{\testtrue}
27 \newif\ifmultiple\multiplefalse
28 \DeclareOption{multiple}{\multipletrue}
29 \DeclareOption*{\PassOptionsToPackage{\CurrentOption}{problem}}
30 \ProcessOptions
```

Then we make sure that the necessary packages are loaded (in the right versions).

```
31 \RequirePackage{keyval}[1997/11/10]
32 \if@hwexam@mh@\RequirePackage{hwexam-mh}\fi
33 \RequirePackage{problem}
```

\hwexam@*@kw For multilinguality, we define internal macros for keywords that can be specialized in *.ldf files.

```
34 \AfterBabelLanguage{ngerman}{\input{hwexam-ngerman.ldf}}
35 \AfterBabelLanguage{arabic}{\input{hwexam-arabic.ldf}}
36 \newcommand\hwexam@assignment@kw{Assignment}
37 \newcommand\hwexam@given@kw{Given}
38 \newcommand\hwexam@due@kw{Due}
39 \newcommand\hwexam@testemptypage@kw{This page was intentionally left blank for extra
40   space}%
41 \newcommand\correction@probs@kw{prob.}%
42 \newcommand\correction@pts@kw{total}%
43 \newcommand\correction@reached@kw{reached}%
44 \newcommand\correction@sum@kw{Sum}%
45 \newcommand\correction@grade@kw{grade}%
46 \newcommand\correction@forgrading@kw{To be used for grading, do not write here}
```

5.2   Assignments

Then we set up a counter for problems and make the problem counter inherited from `problem.sty` depend on it. Furthermore, we specialize the `\prob@label` macro to take the assignment counter into account.

```
47 \newcounter{assignment}
48 \numberproblemsin{assignment}
49 \renewcommand\prob@label[1]{\arabic{assignment}.#1}
```

We will prepare the keyval support for the `assignment` environment.

```
50 \srefaddidkey{assig}
51 \addmetakey{assig}{number}
52 \addmetakey*{assig}{title}
53 \addmetakey{assig}{type}
54 \addmetakey{assig}{given}
55 \addmetakey{assig}{due}
56 \addmetakey[false]{assig}{loadmodules}[true]
```

The next three macros are intermediate functions that handle the case gracefully, where the respective token registers are undefined.

The `\given@due` macro prints information about the given and due status of the assignment. Its arguments specify the brackets.

```
57 \newcommand\given@due[2]{%
58 \ifx \inclassig@given\@empty
59 \ifx \assig@given\@empty
60 \ifx \inclassig@due\@empty
61 \ifx \assig@due\@empty% all empty do nothing
62 \else #1%
63 \fi
64 \else #1%
65 \fi
66 \else #1%
67 \fi
68 \else #1%
69 \fi
70 \ifx\inclassig@given\@empty
71 \ifx\assig@given\@empty% do nothing
72 \else \hwexam@given@kw\xspace \assig@given%
73 \fi
74 \else \hwexam@given@kw\xspace \inclassig@given%
75 \fi
76 \ifx \inclassig@due\@empty
77 \ifx \assig@due\@empty% do nothing
78 \else
79 \ifx \inclassig@given\@empty
80 \ifx \assig@given\@empty% do nothing
81 \else ,~%
82 \fi
83 \else ,~%
84 \fi
85 \fi
86 \else
87 \ifx \inclassig@given\@empty
88 \ifx \assig@given\@empty% do nothing
89 \else ,~%
90 \fi
91 \else ,~%
92 \fi
93 \fi
94 \ifx \inclassig@due\@empty
95 \ifx \assig@due\@empty% do nothing
96 \else \hwexam@due@kw\xspace \assig@due%
97 \fi
98 \else \hwexam@due@kw\xspace \inclassig@due%
99 \fi
100 \ifx \inclassig@given\@empty
101 \ifx \assig@given\@empty
102 \ifx \inclassig@due\@empty
103 \ifx \assig@due\@empty% all empty do nothing
```

```
104 \else #2%
105 \fi
106 \else #2%
107 \fi
108 \else #2%
109 \fi
110 \else #2%
111 \fi
112 }
```

\assignment@title   This macro prints the title of an assignment, the local title is overwritten, if there is one from the \includeassignment. \assignment@title takes three arguments the first is the fallback when no title is given at all, the second and third go around the title, if one is given.

```
113 \newcommand\assignment@title[3]
114 {\ifx\inclassig@title\@empty% if there is no outside title
115 \ifx\assig@title\@empty{#1}\else{#2\assig@title{#3}}\fi
116 \else{#2}\inclassig@title{#3}\fi}% else show the outside title
```

\assignment@number   Like \assignment@title only for the number, and no around part.

```
117 \newcommand\assignment@number%
118 {\ifx\inclassig@number\@empty% if there is no outside number
119 \ifx\assig@number\@empty\else\assig@number\fi
120 \else\inclassig@number\fi}% else show the outside number
```

With them, we can define the central assignment environment. This has two forms (separated by \ifmultiple) in one we make a title block for an assignment sheet, and in the other we make a section heading and add it to the table of contents. We first define an assignment counter

assignment   For the assignment environment we delegate the work to the @assignment environment that depends on whether multiple option is given.

```
121 \newenvironment{assignment}[1][]{\metasetkeys{assig}{#1}\sref@target%
122 \edef\@@num{\assignment@number}%
123 \ifx\@@num\@empty\stepcounter{assignment}\else\setcounter{assignment}{\@@num}\fi%
124 \setcounter{problem}{0}%
125 \def\current@section@level{\document@hwexamtype}%
126 \sref@label@id{\document@hwexamtype \thesection}%
127 \begin{@assignment}}
128 {\end{@assignment}}
```

In the multi-assignment case we just use the omdoc environment for suitable sectioning.

```
129 \def\@@asstitle{\protect\document@hwexamtype~\arabic{assignment}%
130 \assignment@title{}{\;(}{)\;} -- \given@due{}{}}
131 \ifmultiple
132 \newenvironment{@assignment}%
133 {\ifx\assig@loadmodules\@true
134 \begin{omgroup}[loadmodules]{\@@asstitle}
```

```
135 \else
136 \begin{omgroup}{\@@asstitle}
137 \fi}
138 {\end{omgroup}}
```

for the single-page case we make a title block from the same components.

```
139 \else
140 \newenvironment{@assignment}
141 {\begin{center}\bf
142 \Large\@title\strut\\
143 \document@hwexamtype~\arabic{assignment}\assignment@title{\;}{:\;}{\\}%
144 \large\given@due{--\;}{\;--}
145 \end{center}}
146 {}
147 \fi% multiple
```

5.3 Including Assignments

\in*assignment   This macro is essentially a glorified \include statement, it just sets some internal macros first that overwrite the local points Importantly, it resets the inclassig keys after the input.

```
148 \addmetakey{inclassig}{number}
149 \addmetakey*{inclassig}{title}
150 \addmetakey{inclassig}{type}
151 \addmetakey{inclassig}{given}
152 \addmetakey{inclassig}{due}
153 \addmetakey{inclassig}{mhrepos}
154 \clear@inclassig@keys%initially
155 \newcommand\includeassignment[2][]{\metasetkeys{inclassig}{#1}%
156 \include{#2}\clear@inclassig@keys}
157 \newcommand\inputassignment[2][]{\metasetkeys{inclassig}{#1}%
158 \input{#2}\clear@inclassig@keys}
```

5.4 Typesetting Exams

\quizheading

```
159 \addmetakey{quizheading}{tas}
160 \newcommand\quizheading[1]{\def\@tas{#1}%
161 \large\noindent NAME: \hspace{8cm}  MAILBOX:\\[2ex]%
162 \ifx\@tas\@empty\else%
163 \noindent TA: \@for\@I:=\@tas\do{{\Large$\Box$}\@I\hspace*{1em}}\\[2ex]\fi}
```

\testheading

```
164 \addmetakey{testheading}{min}
165 \addmetakey{testheading}{duration}
166 \addmetakey{testheading}{reqpts}
167 \newenvironment{testheading}[1][]{\metasetkeys{testheading}{#1}
168 {\noindent\large{}Name: \hfill Matriculation Number:\hspace*{2cm}\strut\\[1ex]
169 \begin{center}\Large\textbf{\@title}\\[1ex]\large\@date\\[3ex]\end{center}
```

```
170 {\textbf{You have
171 \ifx\testheading@duration\@empty\testheading@min minutes\else\testheading@duration\fi
172 (sharp) for the test}};\\ Write the solutions to the sheet.}\par\noindent
173
174 \newcount\check@time\check@time=\testheading@min
175 \advance\check@time by -\theassignment@totalmin
176 The estimated time for solving this exam is {\theassignment@totalmin} minutes,
177 leaving you {\the\check@time} minutes for revising your exam.
178
179 \newcount\bonus@pts\bonus@pts=\theassignment@totalpts
180 \advance\bonus@pts by -\testheading@reqpts
181 You can reach {\theassignment@totalpts} points if you solve all problems. You will only need
182 {\testheading@reqpts} points for a perfect score, i.e.\ {\the\bonus@pts} points are
183 bonus points. \vfill
184 \begin{center}
185   {\Large\em
186 %  You have ample time, so take it slow and avoid rushing to mistakes!\\[2ex]
187   Different problems test different skills and knowledge, so do not get stuck on
188   one problem.}\vfill\par\correction@table \\[3ex]
189 \end{center}}
190 {\newpage}
```

\testspace

```
191 \newcommand\testspace[1]{\iftest\vspace*{#1}\fi}
```

\testnewpage

```
192 \newcommand\testnewpage{\iftest\newpage\fi}
```

\testemptypage

```
193 \newcommand\testemptypage[1][]{\iftest\begin{center}\hwexam@testemptypage@kw\end{center}\vfill\
```

\@problem   This macro acts on a problem's record in the `*.aux` file. Here we redefine it (it was defined to do nothing in `problem.sty`) to generate the correction table.

```
194 \renewcommand\@problem[3]{\stepcounter{assignment@probs}
195 \def\@@pts{#2}\ifx\@@pts\@empty\else\addtocounter{assignment@totalpts}{#2}\fi
196 \def\@@min{#3}\ifx\@@min\@empty\else\addtocounter{assignment@totalmin}{#3}\fi
197 \xdef\correction@probs{\correction@probs & #1}%
198 \xdef\correction@pts{\correction@pts & #2}
199 \xdef\correction@reached{\correction@reached &}}
```

\correction@table   This macro generates the correction table

```
200 \newcounter{assignment@probs}
201 \newcounter{assignment@totalpts}
202 \newcounter{assignment@totalmin}
203 \def\correction@probs{\correction@probs@kw}%
204 \def\correction@pts{\correction@pts@kw}%
205 \def\correction@reached{\correction@reached@kw}%
206 \def\after@correction@table{}%
207 \stepcounter{assignment@probs}
```

```
208 \newcommand\correction@table{\begin{tabular}{|l|*{\theassignment@probs}{c|}|l|}\hline%
209 &\multicolumn{\theassignment@probs}{c||}%|
210 {\footnotesize\correction@forgrading@kw} &\\\hline
211 \correction@probs & \correction@sum@kw & \correction@grade@kw\\\hline
212 \correction@pts &\theassignment@totalpts & \\\hline
213 \correction@reached & & \\[.7cm]\hline
214 \end{tabular}
215 \ifx\after@correction@table\@empty\else\strut\par\noindent\after@correction@table\fi}
216 ⟨/package⟩
```

5.5 Leftovers

at some point, we may want to reactivate the logos font, then we use

```
here we define the logos that characterize the assignment
\font\bierfont=../assignments/bierglas
\font\denkerfont=../assignments/denker
\font\uhrfont=../assignments/uhr
\font\warnschildfont=../assignments/achtung

\newcommand\bierglas{{\bierfont\char65}}
\newcommand\denker{{\denkerfont\char65}}
\newcommand\uhr{{\uhrfont\char65}}
\newcommand\warnschild{{\warnschildfont\char 65}}
\newcommand\hardA{\warnschild}
\newcommand\longA{\uhr}
\newcommand\thinkA{\denker}
\newcommand\discussA{\bierglas}
```

Index

Numbers written in italic refer to the page where the corresponding entry is described; numbers underlined refer to the code line of the definition; numbers in roman refer to the code lines where the entry is used.

Change History

References

[Koh16a] Michael Kohlhase. *metakeys.sty: A generic framework for extensible Metadata in LATEX*. Tech. rep. Comprehensive TEX Archive Network (CTAN), 2016. URL: http://mirror.ctan.org/macros/latex/contrib/stex/sty/metakeys/metakeys.pdf.

[Koh16b] Michael Kohlhase. *omdoc.sty/cls: Semantic Markup for Open Mathematical Documents in LATEX*. Tech. rep. Comprehensive TEX Archive Network (CTAN), 2016. URL: http://mirror.ctan.org/macros/latex/contrib/stex/sty/omdoc/omdoc.pdf.

[Koh16c] Michael Kohlhase. *problem.sty: An Infrastructure for formatting Problems*. Tech. rep. Comprehensive TEX Archive Network (CTAN), 2016. URL: http://mirror.ctan.org/macros/latex/contrib/stex/sty/problem/problem.pdf.

[Koh16d] Michael Kohlhase. *sref.sty: Semantic Crossreferencing in LATEX*. Tech. rep. Comprehensive TEX Archive Network (CTAN), 2016. URL: http://mirror.ctan.org/macros/latex/contrib/stex/sty/sref/sref.pdf.

[sTeX] *KWARC/sTeX*. URL: https://github.com/KWARC/sTeX (visited on 05/15/2015).