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

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

February 16, 2021

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

1	Inti	roduction	2								
2	The User Interface										
	2.1	Package and Class Options	2								
	2.2	Assignments	2								
	2.3	Typesetting Exams	2								
	2.4		3								
3	Lim	itations	4								
4	Imp	plementation: The hwexam Class	5								
		Class Options	5								
5	Implementation: The hwexam Package										
	5.1	Package Options	5								
	5.2	Assignments	6								
	5.3	Including Assignments	9								
	5.4	Typesetting Exams	9								
	5.5		11								

^{*}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 [Koh20d]. 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

The hwexam class takes the mh option that turns on MathHub support; see [Koh20a] The hwexam package and class take the options solutions, notes, hints, gnotes, pts, min, and boxed that are just passed on to the problems package (cf. its documentation for a description of the intended behavior).

showmeta

mh

If the **showmeta** option is set, then the metadata keys are shown (see [Koh20b] for details and customization options).

The hwexam class additionally accepts the options report, book, chapter, part, and showignores, of the omdoc package [Koh20c] on which it is based and passes them on to that. For the extrefs option see [Koh20e].

2.2 Assignments

assignment number

This package supplies the assignment environment that groups problems into 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 (for the assignment title; this is referenced in the title of the assignment sheet), type (for the assignment type; e.g. "quiz", or "homework"), given (for the date the assignment was given), and due (for the date the assignment is due).

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

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

min

reqpts 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)

February 16, 2021

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.

	To be used for grading, do not write here								
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									

good luck

Example 1: A generated test heading.

2.4 Including Assignments

\inputassignment

The \inputassignment macro can be used to input 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.

number title type

given due

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 \langle / \mathsf{cls} \rangle
```

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 \(\paramox\) newif\if@hwexam@mh@\@hwexam@mh@false
```

```
24 \DeclareOption{mh}{\@hwexam@mh@true}
             25 \neq 25 
             27 \neq 27 
             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 \newcommand\hwexam@assignment@kw{Assignment}
             35 \newcommand\hwexam@given@kw{Given}
             36 \newcommand\hwexam@due@kw{Due}
             37 \newcommand\hwexam@testemptypage@kw{This page was intentionally left blank for extra
                 space}%
             39 \newcommand\correction@probs@kw{prob.}%
             40 \newcommand\correction@pts@kw{total}%
             41 \newcommand\correction@reached@kw{reached}%
             42 \newcommand\correction@sum@kw{Sum}%
             43 \newcommand\correction@grade@kw{grade}%
             44 \newcommand\correction@forgrading@kw{To be used for grading, do not write here}
                For the other languages, we set up triggers
             45 \AfterBabelLanguage{ngerman}{\input{hwexam-ngerman.ldf}}
             46 \AfterBabelLanguage{finnish}{\input{hwexam-finnish.ldf}}
             47 \AfterBabelLanguage{french}{\input{hwexam-french.ldf}}
             48 \AfterBabelLanguage{russian}{\input{hwexam-russian.ldf}}
             5.2
                    Assignments
             Then we set up a counter for problems and make the problem counter inherited
```

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.

```
50 \numberproblemsin{assignment}
51 \renewcommand\prob@label[1]{\arabic{assignment}.#1}
We will prepare the keyval support for the assignment environment.
52 \srefaddidkey{assig}
53 \addmetakey{assig}{number}
54 \addmetakey*{assig}{title}
55 \addmetakey*{assig}{type}
56 \addmetakey{assig}{given}
57 \addmetakey{assig}{due}
58 \addmetakey[false]{assig}{loadmodules}[true]
```

49 \newcounter{assignment}

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.

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

```
104 \ifx \inclassig@due\@empty
105 \ifx \assig@due\@empty% all empty do nothing
106 \else #2%
107\fi
108 \else #2%
109 \fi
110 \else #2%
111 \fi
112 \else #2%
113 \fi
114 }
```

\assignment@title

This macro prints the title of an assignment, the local title is overwritten, if there is one from the \inputassignment. \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.

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

118 $\left\{ 2\right\} \in \$ else show the outside title

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

- 119 \newcommand\assignment@number%
- 120 {\ifx\inclassig@number\@empty% if there is no outside number
- 121 \ifx\assig@number\@empty\else\assig@number\fi
- 122 \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.

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

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

- 131 \def\@@asstitle{\protect\document@hwexamtype~\arabic{assignment}%
- 132 \assignment@title{} ${\;()}$ -- \given@due{}{}}
- 133 \ifmultiple
- 134 \newenvironment{@assignment}%

```
135 {\ifx\assig@loadmodules\@true
136 \begin{omgroup}[loadmodules]{\@@asstitle}
137 \else
138 \begin{omgroup}{\@@asstitle}
139 \fi}
140 {\end{omgroup}}
for the single-page case we make a title block from the same components.
141 \else
142 \newenvironment{@assignment}
143 {\begin{center}\bf
144 \Large\@title\strut\\
145 \document@hwexamtype~\arabic{assignment}\assignment@title{\;}{:\;}{\\}%
146 \large\given@due{--\;}{\;--}
147 \end{center}}
148 {}
149 \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.

```
150 \addmetakey{inclassig}{number}
151 \addmetakey*{inclassig}{title}
152 \addmetakey{inclassig}{type}
153 \addmetakey{inclassig}{given}
154 \addmetakey{inclassig}{due}
155 \addmetakey{inclassig}{mhrepos}
156 \clear@inclassig@keys%initially
157 \newcommand\inputassignment[2][]{\metasetkeys{inclassig}{#1}%
158 \edef\temp@path{#2}
159 \if@iswindows@\path@to@windows\temp@path\fi
160 \input{\temp@path}\clear@inclassig@keys}
161 \newcommand\includeassignment[2][]{\newpage\inputassignment[#1]{#2}}
```

5.4 Typesetting Exams

169 \addmetakey{testheading}{reqpts}

```
\quizheading
```

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

167 \addmetakey{testheading}{min}
168 \addmetakey{testheading}{duration}
```

```
170 \newenvironment{testheading}[1][]{\metasetkeys{testheading}{#1}
                   171 {\noindent\large{}Name: \hfill Matriculation Number:\hspace*{2cm}\strut\\[1ex]
                  172 \end{center} \arge\textbf{\center} \arge\center} \arge\textbf{\center} \arge\center} \arge\center
                  173 {\textbf{You have
                  174 \ifx\testheading@duration\Gempty\testheading@min minutes\else\testheading@duration\fi
                  175 (sharp) for the test}};\\ Write the solutions to the sheet.}\par\noindent
                  176
                  177 \newcount\check@time\check@time=\testheading@min
                  178 \advance\check@time by -\theassignment@totalmin
                  179 The estimated time for solving this exam is {\theassignment@totalmin} minutes,
                  180 leaving you {\the\check@time} minutes for revising your exam.
                  182 \newcount\bonus@pts\bonus@pts=\theassignment@totalpts
                  183 \advance\bonus@pts by -\testheading@reqpts
                   184 You can reach {\theassignment@totalpts} points if you solve all problems. You will only need
                  185 {\testheading@reqpts} points for a perfect score, i.e.\ {\the\bonus@pts} points are
                  186 \text{ bonus points. } \forall \text{vfill}
                  187 \begin{center}
                  188 {\Large\em
                  189 % You have ample time, so take it slow and avoid rushing to mistakes!\\[2ex]
                  190 Different problems test different skills and knowledge, so do not get stuck on
                  191 one problem.}\vfill\par\resizebox{\textwidth}{!}{\correction@table}\\[3ex]
                  192 \end{center}}
                  193 {\newpage}
       \testspace
                  194 \newcommand\testspace[1] {\iftest\vspace*{#1}\fi}
     \testnewpage
                   195 \newcommand\testnewpage{\iftest\newpage\fi}
   \testemptypage
                  196 \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.
                   197 \renewcommand\@problem[3] {\stepcounter{assignment@probs}
                   198 \def\@Qpts{#2}\ifx\@Qpts\@empty\else\addtocounter{assignment@totalpts}{#2}\fi
                   199 \def\@@min{#3}\ifx\@@min\@empty\else\addtocounter{assignment@totalmin}{#3}\fi
                  200 \xdef\correction@probs{\correction@probs & #1}%
                  201 \xdef\correction@pts{\correction@pts & #2}
                   202 \xdef\correction@reached{\correction@reached &}}
\correction@table This macro generates the correction table
                  203 \newcounter{assignment@probs}
                   204 \newcounter{assignment@totalpts}
                  205 \newcounter{assignment@totalmin}
                  206 \def\correction@probs{\correction@probs@kw}%
                  207 \def\correction@pts{\correction@pts@kw}%
```

```
208 \def\correction@reached{\correction@reached@kw}%
209 \def\after@correction@table{}%
210 \stepcounter{assignment@probs}
211 \newcommand\correction@table{\resizebox{\textwidth}{!}{%}
212 \begin{tabular}{|1|*{\theassignment@probs}{c|}|1|}\hline%
213 &\multicolumn{\theassignment@probs}{c|}%|
214 {\footnotesize\correction@forgrading@kw} &\\hline
215 \correction@probs & \correction@sum@kw & \correction@grade@kw\\hline
216 \correction@probs &\theassignment@totalpts & \\hline
217 \correction@reached & & \\[.7cm]\hline
218 \end{tabular}}
219 \ifx\after@correction@table\@empty\else\strut\par\noindent\after@correction@table\fi}
220 \/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\uhrf{\uhrfont\char65}} \newcommand\warnschildf{\warnschildfont\char 65}} \newcommand\hardA{\warnschild} \newcommand\longA{\uhr} \newcommand\hinkA{\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.

LATEXML,

5

Change History

v0.9 General: First Version with Documentation	assignment to hwexam to avoid name clashes with existing assignment.cls on CTAN
v0.9a General: more semantic headers for exams	v1.0 General: adding MathHub support
v0.9b General: adding assignment.cls . 1 v0.9c General: renaming from	v1.1 General: moving MathHub support out to separate package
References	

- [Koh20a] Michael Kohlhase. MathHub Support for sT_EX. Tech. rep. 2020. URL: https://github.com/sLaTeX/sTeX/raw/master/sty/mathhub/ mathhub.pdf.
- [Koh20b] Michael Kohlhase. metakeys.sty: A generic framework for extensible Metadata in LATEX. Tech. rep. 2020. URL: https://github.com/ sLaTeX/sTeX/raw/master/sty/metakeys/metakeys.pdf.
- [Koh20c] Michael Kohlhase. omdoc.sty/cls: Semantic Markup for Open Mathematical Documents in LATEX. Tech. rep. 2020. URL: https://github. com/sLaTeX/sTeX/raw/master/sty/omdoc/omdoc.pdf.
- [Koh20d] Michael Kohlhase. problem.sty: An Infrastructure for formatting Problems. Self-documenting LATEX package. 2020. URL: https:// github.com/sLaTeX/sTeX/raw/master/sty/problem/problem.pdf.
- [Koh20e] Michael Kohlhase. sref.sty: Semantic Crossreferencing in LATEX. Self-documenting LATEX package. 2020. URL: https://github.com/ sLaTeX/sTeX/raw/master/sty/sref/sref.pdf.
- [sTeX]sTeX: A semantic Extension of TeX/LaTeX. URL: https://github. com/sLaTeX/sTeX (visited on 05/11/2020).