

The homework class and style*

Matt Bauman
mbauman@gmail.com

February 12, 2011

Abstract

This package contains both a class and a style designed to simplify the authoring of schoolwork, homework and assignments. They may be used independently of each other; the class provides some slight modifications to the `article` class, while the style adds commonly used packages and functionalities.

Contents

1	Introduction	1
2	The homework class	2
2.1	Arbitrary section numbering	2
2.2	Class options	2
2.3	Implementation	2
2.3.1	Setup	2
2.3.2	Class option handling	3
2.3.3	Arbitrary section numbering	3
2.3.4	Document titling	4
3	The homework style	4
3.1	Implementation	5

1 Introduction

The `homework` package aims to put together a thorough and modern solution for the authoring of schoolwork, homework and assignments. I (Matt Bauman) am by no means a \LaTeX expert. I have, however, made my best effort in creating a package that uses the current best-practices in \LaTeX authoring. This means that `homework` relies on some relatively new packages (like $\epsilon\text{-TeX}$) that may not be available on older systems and some web-based compilers.

*This document corresponds to `homework` v0.1, dated 2011/02/12.

2 The homework class

The `homework` class provides minor enhancements and modifications to the `article` base class. Only alterations which cannot be reliably achieved across a variety of document classes are implemented here.

2.1 Arbitrary section numbering

Most notably, the `homework` class allows for the user to define arbitrary section numbers. As homework assignments are very closely tied to the *number* of the problem, relying on automatic sequential numbering can be problematic. Additionally, problem numbers are not always sequential or even sensible. Thus, the `homework` class augments the standard `\section`, `\subsection`, etc, syntax to optionally specify its number. For simplicity, I will describe everything in terms of `\section`, but this modification applies to all sectioning commands.

The optional argument of the original syntax `\section[toc-name]{sec-name}` is changed to allow a prefix `[number|toc-name]`. Recall that the `toc-name` is how the section will be reported to the table of contents and headers, and that when omitted, it is the same as `sec-name`. This addition attempts to be as compatible as possible with the original syntax. If a `|` character appears within the optional argument, then everything before it is considered the ‘number’ and everything after is the section name for the table of contents. Note that a `|` character may be ‘hidden’ by enclosing it within a `{}` group, in which case it is no longer considered the separator.

Note that ‘empty’ parts of the optional argument are handled differently, depending upon which part was omitted. If the `toc-name` is omitted, e.g., `\section[number|]{sec-name}`, then the section name is used as the name for the table of contents. If, however, the `number` is omitted and the `|` remains, e.g., `\section[|toc-name]{sec-name}` then the section number is set to be empty.

2.2 Class options

In addition to the standard options provided by `article`, the `homework` class adds a `screen/print` option pair. These mutually exclusive options do not have very much functionality in the class currently. They do, however, change the default sidedness of the `article` class (`screen` defaults to `oneside` and `print` defaults to `twoside`). In addition, the `homework` style uses this switch to configure some options for the `hyperref` package.

2.3 Implementation

1 `*class`

2.3.1 Setup

Load `fix-cm` first, as per its instructions. This fixes some bugs in the Computer Modern font and allows for continuous scaling.

```
2 \RequirePackage{fix-cm}
```

Similarly, the font `rsfs` must be modified to support continuous font scaling. See <http://tex.stackexchange.com/q/10698> for details; thanks to user [Leo Liu](#) for this solution. (This could potentially be split off into a simple `fix-rsfs` package.)

```
3 \DeclareFontFamily{U}{rsfs}{\skewchar\font127 }
4 \DeclareFontShape{U}{rsfs}{m}{n}{ % Allow continuous sizing
5   <-6> rsfs5
6   <6-8> rsfs7
7   <8-> rsfs10
8 }{}
```

The `etoolbox` package is required for some of the operations within this class file, including `\newtoggle`, `\ifcsundef` and `\ifstrempy`.

```
9 \RequirePackage{etoolbox}
```

2.3.2 Class option handling

```
10 \newcommand{\hw@sidedness}[1]{\def\hw@side{#1side}}
11 \newtoggle{hw@print}
12
13 \DeclareOption{print}{\toggletrue{hw@print} \hw@sidedness{two}}
14 \DeclareOption{screen}{\togglefalse{hw@print} \hw@sidedness{one}}
15 \DeclareOption{oneside}{\hw@sidedness{one}}
16 \DeclareOption{twoside}{\hw@sidedness{two}}
17
18 \DeclareOption*{\PassOptionsToClass{\CurrentOption}{article}}
19
20 \ExecuteOptions{11pt,screen}
21 \ProcessOptions\relax
22
23 \LoadClass[\hw@side]{article}
```

2.3.3 Arbitrary section numbering

To implement this, first, save the kernel `\@sect` command as `\@@sect`.

```
24 \let\@@sect\@sect
```

Then, redefine `\@sect` to call the function that will handle the parsing and implementation of the new syntax. Add two `|` at the end of the optional argument to ensure that there will *always* be at least three parts separated by `|`.

```
25 \def\@sect#1#2#3#4#5#6[#7]#8{ %
26   \hw@sectsplit{#1}{#2}{#3}{#4}{#5}{#6}}[#7|]{#8}
27 }
```

The `\hw@sectsplit` macro is the meat of the implementation of arbitrary numbering. It parses the optional argument into three parts, `#3`, `#4`, and `#5`.

```
28 \def\hw@sectsplit#1#2[#3|#4|#5]#6{ %
```

As the `\thesection` (or `\thesubsection`, etc, but for simplicity, I will describe this in terms of `\section`) macro is overwritten whenever a custom number is used, we need to ensure that the original value is saved. The first time a sectioning command is called, we save this value into `\hw@theorigsection`. Note that this has the side-effect that the user may not redefine `\thesection` in the middle of the document.

```
29 \ifcsundef{hw@theorig#1}
30   {\expandafter\edef\csname hw@theorig#1\endcsname %
31     {\expandafter\expandonce\csname the#1\endcsname}}
32   {\relax}
```

Now we must parse the optional argument. Argument `#5` simply absorbs any extra `|`s. If it is empty, then that means that there were no `|`s in the input, and only a `toc-name` was specified. In this case, simply ensure that `\thesection` is defined as its original definition and call the kernel's `\@sect` using the defined `toc-name`.

```
33 \ifstrempy{#4#5}
34 {
35   \expandafter\edef\csname the#1\endcsname %
36     {\expandafter\noexpand\csname hw@theorig#1\endcsname}
37   \@sect{#1}#2[#{#3}]{#6}
38 }
```

If, however, argument `#5` was not empty, then the user is calling the new custom syntax. We define the `\thesection` as argument `#3` and then call the kernel's `\@sect` command. If argument `#4` is empty, use the default `toc-name`. Otherwise, use the input provided by the user in argument `#4`.

```
39 {
40   \expandafter\edef\csname the#1\endcsname{#3}
41   \ifstrempy{#4}
42     {\@sect{#1}#2[#{#6}]{#6}}
43     {\@sect{#1}#2[#{#4}]{#6}}
44   }
45 }
```

2.3.4 Document titling

The homework class also provides a few convenience macros to simplify document titling.

```
46 \newcommand*{\hwClass}[1]{\def\@hwClass{#1}}
47 \newcommand*{\hwTitle}[1]{\def\@hwTitle{#1}}
48 \title{\textbf{\@hwClass:} \@hwTitle}
49 \</class>
```

3 The homework style

Put text here.

3.1 Implementation

```
50 <*package>

Here follows the source:
51 \usepackage{fixltx2e}
52 % Use utf-8 encoding for foreign characters
53 \usepackage[T1]{fontenc}
54 \usepackage[utf8]{inputenc}
55 \usepackage[scaled=.86]{beramono}
56 \usepackage{textcomp}
57 % Use microtype, but with half the expansion and protruding punctuation
58 \usepackage[stretch=10,protrusion=true]{microtype}
59
60 % Math stuffs
61 \usepackage{amsmath,amsthm,amssymb}
62 \usepackage{mathtools}
63 \usepackage{dsfont} % \mathds{R} for reals, etc
64 \usepackage{mathrsfs} % \mathscr for scripts
65 \usepackage{xfrac} % \sfrac{1}{2} for slanted fractions
66 \usepackage{empheq}
67 \newcommand{\sch@swap}[2]{\let\sch@tmp#1 \let#1#2 \let#2\sch@tmp}
68 \sch@swap{\theta}{\vartheta}
69 \sch@swap{\phi}{\varphi}
70 \sch@swap{\epsilon}{\varepsilon}
71
72 % Graphics and colors
73 \usepackage[svgnames]{xcolor}
74 \usepackage{graphicx}
75
76 % amazing unit rendering with si{\micro{}}A/cm^2, SI{3}{\meters\per\second}
77 \usepackage{siunitx}
78 \sisetup{per-mode = symbol} % use units in 'm/s' format
79 % And good chemical formula rendering
80 \usepackage[version=3]{mhchem}
81
82 % Figure handling
83 \usepackage{float} % Allow "unfloating" with the H placement specifier
84 \usepackage{wrapfig}
85 % \floatstyle{boxed}
86 % \restylefloat{figure}
87 \usepackage[small,labelfont=bf]{caption}
88 \DeclareCaptionFont{singlespacing}{\setstretch{1}}
89 \captionsetup{font=singlespacing}
90
91 \usepackage{placeins} % Allow \FloatBarrier
92
93 % Package for including code in the document
94 \usepackage{listings}
95 % For faster processing, load Matlab syntax for listings
96 \lstloadlanguages{Matlab}
```

```

97 \newcommand*{\matlabuserfunctions}[1]{
98   \lstset{language=Matlab, morekeywords=[3]{#1}} }
99 \lstset{language=Matlab,
100         frame=single,
101         basicstyle=\footnotesize\ttfamily,
102         keywordstyle=[1]\color{Blue}\bfseries,
103         keywordstyle=[2]\color{Purple},
104         keywordstyle=[3]\color{Blue}\underbar,
105         identifierstyle=,
106         commentstyle=\footnotesize\ttfamily\itshape\color{Green},
107         stringstyle=\color{Purple},
108         showstringspaces=false,
109         tabsize=5,
110         % Put standard MATLAB functions not included in the default
111         % language here
112         morekeywords={xlim,ylim,var,alpha,factorial,poissrnd,normpdf,normcdf},
113         % Put MATLAB function parameters here
114         morekeywords=[2]{on, off, interp},
115         % Put user defined functions here
116         % morekeywords=[3]{},
117         morecomment=[\]\color{Blue}}{...},
118         numbers=left,
119         firstnumber=1,
120         numberstyle=\footnotesize\color{Blue},
121         stepnumber=5
122     }
123 \newcommand*{\matlabscript}[2]
124   {\begin{itemize}\item[]\lstinputlisting[caption={\texttt{#1.m}. #2},label={lst:#1}]{#1.m}\end{itemize}}
125
126 \usepackage[marginpar]{todo}
127
128 % \iftoggle{hw@print}
129 %   {\usepackage{hyperref}}
130 \usepackage[colorlinks,linkcolor=blue]{hyperref}
131 \newcommand*{\magicref}[2]{\hyperref[#2]{#1 \ref{#2}}}
132
133
134 \usepackage{tikz}
135 \usepackage{pgfplots}
136 \pgfplotsset{compat=1.4}
137 \end{package}

```