The homework class and style*

Matt Bauman
mbauman@gmail.com

February 12, 2011

Abstract

This package contains a 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

Put text here.

2 The homework class

The homework class is provides minor enhancements and modifications to the article base class which cannot be reliably achieved within the style.

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

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
```

```
  <-6> rsfs5
  <6-8> rsfs7
  <8-> rsfs10
}{}
```

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

```
\RequirePackage{etoolbox}
```

2.3.2 Class option handling

```
\newcommand{\hw@sidedness}[1]{\def\hw@side{#1side}}
\newtoggle{hw@print}

\DeclareOption{print}{\toggletrue{hw@print}   \hw@sidedness{two}}
\DeclareOption{screen}{\togglefalse{hw@print} \hw@sidedness{one}}
\DeclareOption{oneside}{\hw@sidedness{one}}
\DeclareOption{twoside}{\hw@sidedness{two}}

\DeclareOption*{\PassOptionsToClass{\CurrentOption}{article}}

\ExecuteOptions{11pt,screen}
\ProcessOptions\relax

\LoadClass[\hw@side]{article}
```

2.3.3 Arbitrary section numbering

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

```
\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 `|`.

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

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`.

```
\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.

```
  \ifcsundef{hw@theorig#1}
    {\expandafter\edef\csname hw@theorig#1\endcsname %
```

```
      {\expandafter\expandonce\csname the#1\endcsname}}
    {\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`.

```
  \ifstrempty{#4#5}
  {
    \expandafter\edef\csname the#1\endcsname %
      {\expandafter\noexpand\csname hw@theorig#1\endcsname}
    \@@sect{#1}#2[{#3}]{#6}
  }
```

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`.

```
  {
    \expandafter\edef\csname the#1\endcsname{#3}
    \ifstrempty{#4}
      {\@@sect{#1}#2[{#6}]{#6}}
      {\@@sect{#1}#2[{#4}]{#6}}
  }
}
```

2.3.4 Document titling

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

```
\newcommand*{\hwClass}[1]{\def\@hwClass{#1}}
\newcommand*{\hwTitle}[1]{\def\@hwTitle{#1}}
\title{\textbf{\@hwClass:} \@hwTitle}
```

⟨/class⟩

3 The homework style

Put text here.

3.1 Implementation

⟨*package⟩

Here follows the source:

```
\usepackage{fixltx2e}
% Use utf-8 encoding for foreign characters
\usepackage[T1]{fontenc}
```

```
\usepackage[utf8]{inputenc}
\usepackage[scaled=.86]{beramono}
\usepackage{textcomp}
% Use microtype, but with half the expansion and protruding punctuation
\usepackage[stretch=10,protrusion=true]{microtype}

% Math stuffs
\usepackage{amsmath,amsthm,amssymb}
\usepackage{mathtools}
\usepackage{dsfont} % \mathds{R} for reals, etc
\usepackage{mathrsfs} % \mathscr for scripts
\usepackage{xfrac} % \sfrac{1}{2} for slanted fractions
\usepackage{empheq}
\newcommand{\sch@swap}[2]{\let\sch@tmp#1 \let#1#2 \let#2\sch@tmp}
\sch@swap{\theta}{\vartheta}
\sch@swap{\phi}{\varphi}
\sch@swap{\epsilon}{\varepsilon}

% Graphics and colors
\usepackage[svgnames]{xcolor}
\usepackage{graphicx}

% amazing unit rendering with si{\micro{}A/cm^2}, SI{3}{\meters\per\second}
\usepackage{siunitx}
\sisetup{per-mode = symbol} % use units in 'm/s' format
% And good chemical formula rendering
\usepackage[version=3]{mhchem}

% Figure handling
\usepackage{float}    % Allow "unfloating" with the H placement specifier
\usepackage{wrapfig}
% \floatstyle{boxed}
% \restylefloat{figure}
\usepackage[small,labelfont=bf]{caption}
% \DeclareCaptionFont{singlespacing}{\setstretch{1}}
% \captionsetup{font=singlespacing}

\usepackage{placeins} % Allow \FloatBarrier

% Package for including code in the document
\usepackage{listings}
% For faster processing, load Matlab syntax for listings
\lstloadlanguages{Matlab}
\newcommand*{\matlabuserfunctions}[1]{
  \lstset{language=Matlab, morekeywords=[3]{#1}} }
\lstset{language=Matlab,
        frame=single,
        basicstyle=\footnotesize\ttfamily,
        keywordstyle=[1]\color{Blue}\bfseries,
        keywordstyle=[2]\color{Purple},
```

```
        keywordstyle=[3]\color{Blue}\underbar,
        identifierstyle=,
        commentstyle=\footnotesize\ttfamily\itshape\color{Green},
        stringstyle=\color{Purple},
        showstringspaces=false,
        tabsize=5,
        % Put standard MATLAB functions not included in the default
        % language here
        morekeywords={xlim,ylim,var,alpha,factorial,poissrnd,normpdf,normcdf},
        % Put MATLAB function parameters here
        morekeywords=[2]{on, off, interp},
        % Put user defined functions here
        % morekeywords=[3]{},
        morecomment=[l][\color{Blue}]{...},
        numbers=left,
        firstnumber=1,
        numberstyle=\footnotesize\color{Blue},
        stepnumber=5
        }
\newcommand*{\matlabscript}[2]
  {\begin{itemize}\item[]\lstinputlisting[caption={\texttt{#1.m}. #2},label={lst:#1}]{#1.m}\end{i

\usepackage[marginpar]{todo}

% \iftoggle{hw@print}
%   {\usepackage{hyperref}}
\usepackage[colorlinks,linkcolor=blue]{hyperref}
\newcommand*{\magicref}[2]{\hyperref[#2]{#1 \ref{#2}}}


\usepackage{tikz}
\usepackage{pgfplots}
\pgfplotsset{compat=1.4}
```

⟨/package⟩