

The Handout Document Class

Roger Cortesi

July 27, 2008

1 Introduction

The handout document class is an extension of the article document class. It is designed to make producing course handouts easier. The last pages of this document include examples of documents made with the handout class.

2 Required Packages

This document class requires (and makes available) several L^AT_EX packages. These are: ifthen, fancyhdr, lastpage, amssymb, amsmath, mathrsfs, amsfonts, color, and listings.

This class uses the *lastpage* package to display page numbers a 1 of 2 or 2 of 5 etc. This means that the document must be generated **twice** so the *lastpage* package can get an accurate page count.

3 Summary of Features

The class does the following things:

1. Increases the printable area by making the margins smaller. Instructors tend to cram more information on a page than LaTeX considers to be aesthetically pleasing.
2. Make the section, subsection and subsubsection font and space smaller. Again so one can fit more information on a page.
3. Defines a *questions* environment which is an enumerated list that maintains a running count of question numbers even if it is ended and began again later in the document.
4. Defines a page header with fields for course title, handout title, due date, semester and name blanks. Along with the command to fill these fields.

5. Defines a *soln* command to specify the solution to the questions and commands to tell the calls to print the solutions. Solutions are not printed by default. By having the solutions stored in the same document as the questions, you don't have to keep two or more documents synchronized.
6. Redefines the *maketitle* command to show the handout title and due date as a large centered title where the command is called.
7. Defines a *maketitleinst* command to show the handout title and instructor name as a large centered title where the command is called.
8. Defines several new commands and abbreviates some that I find myself using often.
9. Provides a consistent environments for displaying computer code (using the listing package).

4 Usage

4.1 The Bare Minimum

```
\documentclass{handout}
```

4.2 Commands For Setting Page Header Fields

The page header contains the following fields:

1. Course Title
2. Handout Title
3. Semester
4. Due Date

All the above fields are blank by default.

Additionally, the header may contain a couple of blanks for students to write their name, section number and date.

The commands below set these fields.

```
\SetCourseTitle{My Course Title}
\SetHandoutTitle{My Handout Title}
\SetDueDate{My Due Date}
\SetInstructor{My Instructor Name}
\SetSemeste{My Semtester}
\ShowNameBlack
\ShowNameDateBlanks
\ShowAllBlanks
```

The fields above can be displayed in other places in the document using the commands:

```
\CourseTitle
\HandoutTitle
\DueDate
\Instructor
\Semester
\Blank
```

4.3 The Questions Environment

The questions environment can be used for automatic numbering of questions. The question environment maintains a running count of the question numbers even if the environment is stop and started.

```
\begin{questions}
\item This is question 1.
\item This is question 2.
\end{questions}
```

blaa blaa blaa...

```
\begin{questions}
\item This is question 3.
\item This is question 4.
\end{questions}
```

4.4 Solutions

You can specify solutions to the questions in the same documents. By default the handout class will not print the solutions in the documents.

4.4.1 Specifying Solutions

Solutions are specified using the command:

```
\soln{length}{text}
```

The *length* parameter specifies how much vertical whitespace should be displayed when the solution is *not* printed. The *text* argument specifies the text of the solution. For example:

```
\soln{2in}{This is the to the question:  $2+2=4$ }
```

4.4.2 Printing Solutions

There are three methods to tell the handout class to print the solutions.

1. The command *ShowSoln*: Any solutions specified after this command will be printed.

```
\ShowSoln
```

2. As an option in the document class.

```
\documentclass[solution]{handout}
```

3. From the command line when latex is invoked:

```
>latex "\newcommand{\MakeSoln}{ }\input{filename.tex}"
```

This method is particular useful for regenerating the course documents with a batch script, when called with the *-jobname* option to rename the output files. For example:

```
>latex -jobname=soln_filename "\newcommand{\MakeSoln}{ }\input{filename.tex}"
```

Prints the solution of *filename* as *soln_filename*.

4.4.3 Misc. Solution Commands

By default the solutions are printed in blue. The color of the solutions can be specified with the command:

```
\setsolncolor{red}
```

Solutions can be hidden beyond a point in the document with the command:

```
\HideSoln
```

5 Examples

Here is a minimal example of a document that is an assignment with a due date.

```
\documentclass{handout}

\SetInstructor{Roger Cortesi}
\SetCourseTitle{Latex 101}
\SetSemester{Fall 2007}
\SetHandoutTitle{Performing Engineering Calculations}
\SetDueDate{10 Aug 2007}
\ShowAllBlanks

\begin{document}
\maketitle

\begin{questions}
\item What is the meaning of life?

\soln{1in}{42}

\item What is Euler's Equation?

\soln{2in}{ $e^{i\pi} - 1 = 0$ }

\newpage
\item Are the page headers different on the pages which are not the title page?

\soln{1in}{Yes}

\end{questions}

\end{document}
```

5.1 Additional Commands

The commands supplied by this class are.

1. `\mb{<some math text>}` This is an abbreviation for `\mathbf{<some math text>}` command. It can only be used in a math environment.
2. `\mat{ element 1,1 & element 1,2 \ \ element 2,1 & ...etc...}` this is an abbreviation for the `\begin{bmatrix}... \end{bmatrix}` environment.
3. `\pd\{<num>\}{<den>}` creates a partial derivative symbol with the arguments in the numerator and denominator respectively.

5.2 Source Code Display and Formatting

This package creates the environments *matlabCode*, *cCode*, *javaCode*, *fortranCode*, *latexCode*, and *noCode* for type setting source code in those languages. These environments use the *listings* package to provide these services. See the *listings*' packages documentation for more details. The *noCode* environment is the same as the others except no keywords or comments are colored.

All these environments take a second argument. This second argument is passed to a *lstlist* command, it may be left blank. For example to specify a caption for the listing.

```
\begin{matlabCode}{caption={This is a caption}}
    matlab code here
\end{matlabCode}
```

for the listing to be a float, have a captions and a label:

```
\begin{matlabCode}{float , caption={This is a caption} , label=samplecode}
    matlab code here
\end{matlabCode}
```

Other languages can be specified by calling the *lstlisting* environment with a language arguments. For example to specify pascal as language for a code listing.

```
\begin{lstlisting}{language=pascal}
... some pascal source code here...
\end{lstlisting}
```

Handout Title Here

Instructor Name Here

or

Handout Title Here

Due: Due Date Here

Optionally, you may include either of the above titling in your document with the commands `\maketitleinst` or `\maketitle` respectively.

Here are some examples of the added commands. These commands must be run from **within a math environment**:

1. `\mb{x}` applies the `\mathbf{}` command to the argument. This example would result in **x**.
2. `\mat{a & b \ c & d}` would produce $\begin{bmatrix} a & b \\ c & d \end{bmatrix}$.
3. `\pd{X}{t}` produces the partial derivative $\frac{\partial X}{\partial t}$. Arguments can be skipped, `\pd{}{t}` yields $\frac{\partial}{\partial t}$. The second partial derivative would be written as `\pd{^2X}{t^2}` yielding $\frac{\partial^2 X}{\partial t^2}$.

1 Code Listings

The handout class uses the *listings* package. You should make become familiar with it and its options. The handout class defines a series of options to format the code listing in a consistent manor. You can override any of these configurations with calls to *lstlist*.

The handout class also defined the following environments to specify the code listing's language, these environments are:

1. matlabCode
2. cCode
3. javaCode
4. fortranCode
5. latexCode
6. noCode

All these environments take a second argument. This second argument is passed to a *lstlist* command, it may be left blank. For example to specify a caption for the listing.

```
\begin{matlabCode}{caption={This is a caption}}
    matlab code here
\end{matlabCode}
```

for the listing to be a float, have a captions and a label:

```
\begin{matlabCode}{float , caption={This is a caption} , label={samplecode}}
    matlab code here
\end{matlabCode}
```

1.1 MATLAB

The following Latex

```
\begin{matlabCode}{}  
% This function models a mass-spring-damper system  
function xdot = smd(t,x)  
    m = 1;  
    b = 0.05;  
    k = 0.5;  
    xdot = [0 1; -b/m -k/m]*x;  
\end{matlabCode}
```

produces...

```
% This function models a mass-spring-damper system  
function xdot = smd(t,x)  
    m = 1;  
    b = 0.05;  
    k = 0.5;  
    xdot = [0 1; -b/m -k/m]*x;
```