The exam-randomizechoices package

LaTeX package for creating random placed choices in multiple choice environments using the exam document class

Jesse op den Brouw
Department of Electrical Engineering
The Hague University of Applied Sciences
Delft, Netherlamds
J.E.J.opdenBrouw@hhs.nl

Copyright ©2018 Jesse op den Brouw
All rights reserved

Draft: August 13, 2018

This is the user's guide for version 0.1 of the exam-randomizechoices package.

Contents

1 Introduction

1.1 License

This work may be distributed and/or modified under the conditions of the LaTeX Project Public License, either version 1.3 of this license or (at your option) any later version. The latest version of this license is in http://www.latex-project.org/lppl.txt and version 1.3 or later is part of all distributions of LaTeX version 2003/12/01 or later.

This work has the LPPL maintenance status "author-maintained".

This work consists of the files `exam-randomizechoices.sty`, `exam-randomizechoices.tex` and `exam-randomizechoices-doc.tex`

1.2 Acknowledgements

The author wishes to thank the developers of the `exam` document class and the `mcexam` package.

1.3 Warning

This package is experimental, so it could possibly break LaTeX compilation. Use this package with care. Please report any problems to the author.

Please use a recent version of the `exam` document class. This package is tested with version 2.603 which is available in most distributions. Testing with version 2.604$beta $ is planned.

2 Using the `exam` class

The `exam` document class is a powerfull class to create exams with LaTeX. Both open questions and multiple choice questions are supported. For multiple choice questions we can differentiate between enumerated lists or checkbox lists. The class documentation states[1]:

> The file `exam.cls` provides the `exam` document class, which attempts to make it easy for even a LaTeX novice to prepare exams. Specifically, `exam.cls` sets the page layout so that there are one inch margins all around (no matter what size paper you're using) and provides commands that make it easy to format questions, create flexible headers and footers, change the margins, and create grading tables. In more detail:
>
> - The class will automatically format and number the questions, parts of questions, subparts of parts, and subsubparts of subparts.
> - You can include the point value of each question (or part, or subpart, or subsubpart), with your choice of having the point values printed at the beginning of the text of the question, opposite that in the left margin, opposite that in the right margin, or in the right margin opposite the end of the question.

[1] See https://ctan.org/tex-archive/macros/latex/contrib/exam.

- The class will add up the total points for each question (and all of its parts, subparts, and subsubparts) and the total points on each page, and make those totals available in macros.
- You can have the class print a grading table, indexed either by question number or by page number.
- You specify the header in three parts: One part to be left justified, one part to be centered, and one part to be right justified, and one or all of these can be omitted.
- The footer is also specified in three parts: Left justified, centered, and right justified.
- The header and footer for the first page can be different from the ones used on other pages.
- Both headers and footers can contain more than one line. To accommodate headers and footers with several lines, simple commands are provided to enlarge the part of the page devoted to the header and/or footer, and these commands can give one amount of space on the first page and a different amount of space on all other pages.
- Macros are defined to enable you to state the total number of pages in the exam and to change the header and/or footer that appears on the last page of the exam .
- Macros are defined so that the headers and footers can vary depending on whether the current page begins a new question or continues a question that started on an earlier page (and, if one continues onto the current page, to say what the number of that question is). Macros are also defined so that the headers and footers can vary depending on whether a question is complete on the current page or continues on to the next page (and, if one continues, to say what the number of that question is).
- You can have a horizontal rule at the base of the header and/or at the top of the footer.
- The exam can begin with one or more cover pages, which are numbered separately from the main pages of the exam and which can have headers and footers different from the ones in the main pages of the exam.
- You can include solutions in your LaTeX file and have these solutions either printed or ignored (or replaced automatically by space in which the students can write their answers) depending on a single command.

Furthermore, not stated in this excerpt, you can typeset bonus questions with bonus points and grading tables with bonus points.

You can load the `exam` document class the usual way. An example might be:

```
1 \documentclass[a4paper,12pt,addpoints]{exam}
```

which loads the `exam` document class. The paper size is set to A4 (297 mm × 210 mm, 11.7 in × 8.3 in), the document is typeset with 12 points letters and the question points are added.

2.1 Typesetting multiple choice questions

Then, within the document body and between `\begin{questions}` and `\end{questions}`, you enter the questions. Only multiple choice questions are considered here. The `exam` document class provides four types of multiple choice question environments:

`choices` The given choices are typeset in a linear, vertical list. Each given choice is prepended with a label name which can be set to uppercase letter, lowercase letter, Roman numerals (uppercase and lowercase) and the Greek alphabet[2].

`oneparchoices` The given choices are typeset in a linear, horizontal list. Long lists are split over multiple lines. Each given choice is prepended with a label name which can be set to uppercase letter, lowercase letter, Roman numerals (uppercase and lowercase) and the Greek alphabet.

`checkboxes` The given choices are typeset in a linear, vertical list. Each given choice is prepended with a checkbox, which defaults to an open circle.

`oneparcheckboxes` The given choices are typeset in a linear, horizontal list. Long lists are split over multiple lines. Each given choice is prepended with a checkbox, which defaults to an open circle.

Examples of the four environment are given below.

2.2 The `choices` environment

An example of a question with the `choices` environment is:

```
\question[5] What is the result of $1+1$?.

\begin{choices}
\choice 1
\CorrectChoice 2
\choice 3
\choice 4
\end{choices}
```

which is typeset as:

1. (5 points) What is the result of $1 + 1$?.

   A. 1

[2] Provided by the `exam` document class.

B. 2

C. 3

D. 4

2.3 The `oneparchoices` environment

An example of a question with the `oneparchoices` environment is:

```
\question[5] What is the result of $2+2$?..

\begin{oneparchoices}
\choice 1
\choice 2
\choice 3
\CorrectChoice 4
\end{oneparchoices}
```

which is typesets as:

2. (5 points) What is the result of $2+2$?..

A. 1 B. 2 C. 3 D. 4

Furthermore, the `exam` document class provides for two way of typesetting checkbox questions.

2.4 The `checkboxes` environment

An example of a vertically aligned checkbox environment is:

```
\question[5] What is the result of $1+2$?..

\begin{checkboxes}
\choice 1
\choice 2
\CorrectChoice 3
\choice 4
\end{checkboxes}
```

which typesets to:

3. (5 points) What is the result of $1+2$?..

◯ 1

◯ 2

◯ 3

◯ 4

2.5 The `onecheckboxes` environment

The `oneparcheckboxes` environment typesets the choices in al linear, horizontal way. An example is given below:

```
\question[5] What is the result of $1+2$?..

\begin{oneparcheckboxes}
\choice 1
\choice 2
\CorrectChoice 3
\choice 4
\end{oneparcheckboxes}
```

which typesets to:

4. (5 points) What is the result of $1 + 2$?..

   ◯ 1 ◯ 2 ◯ 3 ◯ 4

Other types of questions are not considered here.

Please note the use of the `\choice` and `\CorrectChoice` command. A `\choice` typesets as an item in the list with no special markup. A `\CorrectChoice` typesets an item in the list with no special markup if the `exam` document class option `answers` is *not* given, and typesets with special markup if the class option `answers` is given. This special markup defaults to boldface for the `choices` and `oneparchoices` environments:

```
\question[5] What is the result of $2+2$?..

\begin{choices}
\choice 1
\choice 2
\choice 3
\CorrectChoice 4
\end{choices}
```

which is typesets as:

5. (5 points) What is the result of $2 + 2$?..

   A. 1

   B. 2

   C. 3

   D. 4

Note that item D is typeset in boldface. The `checkboxes` and `oneparcheckboxes` environments use a checkmark:

```
\question[5] What is the result of $1+2$?..

\begin{oneparcheckboxes}
\choice 1
\choice 2
\CorrectChoice 3
\choice 4
\end{oneparcheckboxes}
```

which typesets to:

6. (5 points) What is the result of $1 + 2$?..

◯ 1 ◯ 2 √ **3** ◯ 4

3 The package

3.1 New multiple choices environments

Basically, this package provides the user with four new multiple choices environments:

`randomizechoices` This is the randomizing counterpart of the `choices` environment. It typesets the given items in a random order.

`randomizeoneparchoices` This is the randomizing counterpart of the `oneparchoices` environment. It typesets the given items in a random order.

`randomizecheckboxes` This is the randomizing counterpart of the `checkboxes` environment. It typesets the given items in a random order.

`randomizeoneparcheckboxes` This is the randomizing counterpart of the `oneparcheckboxes` environment. It typesets the given items in a random order.

3.2 Using the new environments

We will discuss the `randomizechoices` environment only. The other environment work alike.

When using the `randomizechoices` environment

```
\question[5] What is the result of $1+1$?.

\begin{randomizechoices}
\choice 1
\CorrectChoice 2
\choice 3
```

```
\choice 4
\end{randomizechoices}
```

which *possibly* is typeset as:

7. (5 points) What is the result of $1 + 1$?.

   A. 1

   B. 3

   C. 2

   D. 4

Here we can see that the resulting output is typeset in another order then the choices are given. We say *possibly* because the output depends on the state of the pseudo random generator. See Section 3.5

3.3 Loading the package

The package is loaded using the well-known `\usepackage` command:

```
\usepackage[ option list ]{exam-randomizechoices}
```

The options in *option list* can be any combination of:

debug This option cause the package to emit a lot of debug messages in the log file. The messages are written to the log by the `\PackageWarning` command. Most IDE's such as TeXMaker will display the messages in the transcript pane. Debug is off by default. There is no `nodebug` option.

random This option globally turns on the randomizing of the choices given for all available typesetting environments. Random is turned on by default.

norandom This option globally turns off the randomizing of the choices for all available typesetting environments. This option is usefull for inspecting the resulting PDF output file with typesetting the choices in the order they were entered.

3.4 Package options

3.5 Seeding rhe pseudo random generator

To get a consistent randomization, you must seed the pseudo random generator with the same seed every time you compile your document. You can set the seed using the `\setrandomizerseed` macro. The macro has a mandatory argument that is a integer between 0 and $2^{31} - 1$, TEX' largest integer. Internallu, the PGF macro `\pgfmathsetseed` is called, and it is flagged that you applied a seed. If you fail to do so, the seeding value is `\time×\year` as stated by the PFG manual[3]. Some LATEX compilers keep track of time by

[3] Version 3.0.1a, page 940.

an integer that holds the seconds since midnight. The integer is incremented every time the time passes a minute boundary. So the scenario can be that you compile your document a couple of times with no apparent differences between runs. But if the time passes a minute boundary, the next time you compile your document you'll see that the environment items have been rearranged.

4 Printing the key table

The package provides the typesetting of a basic key table in vertical direction. Please note that only the environments `randomizechoices`, `randomizeoneparchoices`, `choices` (if overloaded) and `oneparchoices` (if overloaed) can have keys in the key table. The `randomizecheckboxes`, `randomizeoneparcheckboxes`, `checkboxes` and `oneparcheckboxes` environments can't have keys because of the typesetting regime used by the `exam` document class. The `*choices` environments are typeset as lists using `\item` which can be provided with a `\label`. The `*checkboxes` environment are typeset by skips et al. so labeling them would lead to a reference of the current question (or part, or sub part or sub-sub part). Labeling the correct choices (with the `\CorrectChoice` command is automaticly handled by the package.

At the end of the exam, issue the command:

```
1 \printkeytable
```

to print the key table. If an environment can't be labeled, the table entry will contain **??**, otherwise it will contain the used typeseting scheme (which is `\Alph` by default). The key table is typeset using the `tabular` environment, so it can be wrapped in a `table` environment.

Note on the external key table file

The key table is typeset using an external file, called `\jobname.keytable`, where `\jobname` is the name of your LaTeX file you are compiling. Using an external file is far more easy then typesetting it directy from the package. See https://tex.stackexchange.com/questions/367979/latex-foreach-in-tabular-environment why.

The key table file can safely be deleted. It is generated each time the `\printkeytable` command is executed. Edits to the file are lost.

Question	Answer
1	??
2	??
3	??
4	??
5	??
6	??
7	B