

## Creating and automating exams with L<sup>A</sup>T<sub>E</sub>X & friends

Uwe Ziegenhagen

### Abstract

Although L<sup>A</sup>T<sub>E</sub>X is widely used in academia and education only a few teachers use it to prepare exams for their students. In this article we show how the `exam` package can be used to create various exercise types and how exercises can be created randomly using Python.

### 1 Introducing the exam class

The `exam` package [1] is maintained by Philip Hirschhorn; the current version, 2.6, is from November 2017. It supports various question types, described in the well-written manual accompanying the package.

A basic example for an `exam`-based exam sheet can be found in Figure 1 below. It uses `exam` as the document class. Inside the document a `questions` environment is used, with item-like `\question` commands that take the number of achievable points for this exercise as an optional parameter.

For exams in languages other than English, the exam-specific terms can be translated. See Listing 1 for a translation into German.

```
\pointpoints{Punkt}{Punkte}
\bonuspointpoints{Bonuspunkt}{Bonuspunkte}
\renewcommand{\solutiontitle}
  {\noindent \textbf{Lösung:}\enspace}
\chqword{Frage} \chpgword{Seite}
\chpword{Punkte} \chbpword{Bonus Punkte}
\chsword{Erreicht} \chtword{Gesamt}
\hpword{Punkte:} \hsword{Ergebnis:}
\hqword{Aufgabe:} \htword{Summe:}
```

Listing 1: Localization, here for German

The package also allows for defining the layout of headers and footers, separately for the first page and all subsequent pages. An example of the commands and output is shown in Figure 2. Each command

```
\documentclass[12pt]{exam}
\begin{document}\Large
\begin{questions}
\question[10] Who was Albert Einstein?
\question[10] Compute  $(e = m \cdot c^2)$ !
\end{questions}
\end{document}
```

1. (10 points) Who was Albert Einstein?
2. (10 points) Compute  $e = m \cdot c^2$ !

Figure 1: A basic example of `exam`, source and output

```
\pagestyle{headandfoot}
\firstpageheadrule
\runningheadrule
\firstpageheader{<left>}{<center>}
  {John Doe \ Statistics 101 - 2019}
\runningheader{<1>}{<c>}{Statistics 101 - 2019}
\firstpagefooter{\today}{ACME University}
  {\thepage\,/,\, \numpages}
\runningfooter {\today}{ACME University}
  {\thepage\,/,\, \numpages}

\begin{document}\Large
\begin{questions}
\question[10] Who was Albert Einstein?
\question[10] Compute  $(e = m \cdot c^2)$ !
\end{questions}
\end{document}
```

```
<left> <center> John Doe
Statistics 101 - 2019

1. (10 points) Who was Albert Einstein?
2. (10 points) Compute  $e = m \cdot c^2$ !
```

Resulting output (top of page)

```
August 19, 2019 ACME University 1/1
```

Resulting output (bottom of page)

Figure 2: Setting headers and footers

```
\begin{questions}
\question[10] Who was Albert Einstein?
\begin{parts}
\part[1] Where was he born?
\part[4] What has he become famous for?
\begin{subparts}
\subpart[2] What does  $(e=mc^2)$  mean?
\subpart[2] What did he get the Nobel prize for?
\end{subparts}
\end{parts}
\end{questions}
```

```
<left> <center> John Doe
Statistics 101 - 2019

1. (10 points) Who was Albert Einstein?
  (a) (1 point) Where was he born?
  (b) (4 points) What has he become famous for?
    i. (2 points) What does  $e = mc^2$  mean?
    ii. (2 points) What did he get the Nobel prize for?
```

Figure 3: Subdividing questions: `\part` and `\subpart`

has three parameters, for the left, the center, and the right part of the corresponding header/footer.

Questions can be further elaborated. The `exam` package provides the environments `parts`, `subparts`, and `subsubparts`. Inside these environments individual subquestions are added with `\part`, `\subpart` or `\subsubpart`. See Figure 3 for examples.

```

\question Who was not a Beatle?
\begin{choices}
\choice John
\choice Paul
\choice George
\CorrectChoice Benedict
\end{choices}

\question Who was not a Beatle?
\begin{checkboxes}
\choice John
\choice Paul
\choice George
\CorrectChoice Benedict
\end{checkboxes}

```

```

<left> <center> John Doe
Statistics 101 - 2019


---


1. Who was not a Beatle?
A. John
B. Paul
C. George
D. Benedict

2. Who was not a Beatle?
 John
 Paul
 George
 Benedict

```

Figure 4: Examples for `choices` and `checkboxes`

Besides the text-based questions we have seen so far, the `exam` class offers several environments for multiple choice and fill-in questions:

- `choices` for vertical choices using letters
- `checkboxes` for vertical checkboxes
- `oneparcheckboxes` for horizontal checkboxes
- `\fillin[<solution text>]` prints a horizontal line where the students should put their answer.

For multiple choice questions, the correct answer is defined with the `\CorrectChoice` command. To typeset a version of the exam that has the correct answers and solutions highlighted, `answers` is added to the list of class options. See Figures 4 and 5.

To create space for answers, the package not only supports the usual TeX commands (Figure 6), but also “enriched” solution space commands that provide lines, dotted lines or a grid (Figure 7).

To also insert solutions into the exam, one can use the `solution` environment — see Figure 8 — or one of the following environments:

- `solutionorbox`
- `solutionorlines`
- `solutionordottedlines`
- `solutionorgrid`

For the `solutionorgrid` environment an example is shown in Figure 9 which, depending on whether

```

\question Who was not a Beatle?
\begin{oneparcheckboxes}
\choice John
\choice Paul
\choice George
\choice Ringo
\CorrectChoice Benedict
\end{oneparcheckboxes}

\question \fillin[James Bond][11em] has the
\enquote{license to kill}.

<left> <center> John Doe
Statistics 101 - 2019


---


1. Who was not a Beatle?
 John  Paul  George  Ringo  Benedict

2. James Bond has the “license to kill”.

```

Figure 5: `oneparcheckboxes` option and `\fillin` command, with `answers` option set

```

% simple vertical space
\vspace*{<length>}

% vertical space to the end of the page
\vspace*{\stretch{1}}
\newpage

% empty framed box
\makeemptybox{<length>}

% empty framed box to the end of the page
\makeemptybox{\stretch{1}}
\newpage

```

Figure 6: TeX commands to make answer spaces

the class option `answers` is set, either presents a plot of a quadratic function or a grid where the students are to draw the function themselves.

As mentioned earlier, the different question environments take the number of points as optional parameters. To assist with the creation of the grading table, `exam` has commands for producing vertical or horizontal grading tables that are either based on the page or exercise number. Figure 10 shows variations of the `\gradetable` command and example output from `\gradetable[h][questions]`.

## 2 Automating exam

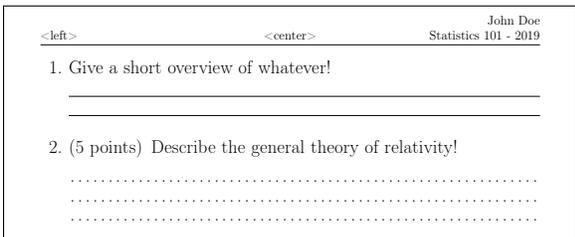
In this section we want to show how exam questions can be created individually for each student, e.g., to prevent cheating. We also use QR codes that are printed behind each exercise, thus generating a teacher-friendly version by eliminating the need to calculate all the individual results herself as a modern smartphone suffices to see the result immediately.

```
\fillwithlines{<length>} % for lines
% Remark: \linefillheight for interline spacing

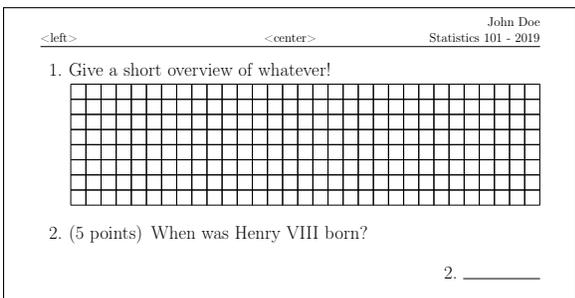
\fillwithdottedlines{<length>} % for dotted lines
% Remark: distance in \dottedlinefillheight

\fillwithgrid{<length>} %
% \setlength{\gridsize}{5mm}
% \setlength{\gridlinewidth}{0.1pt}

\answerline[answer] % for short answers
```



(output from `\fillwithlines` and `\fillwithdottedlines`)



(output from `\fillwithgrid` and `\answerline`)

Figure 7: Examples of enriched answer spaces.

```
\begin{questions}
\question[1] How much does lead (Pb) weigh?

\begin{solution}
Pb weighs \SI{11,342}{\gram\per \centi\meter^3}
\end{solution}

\end{questions}
```

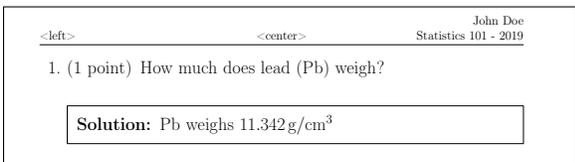


Figure 8: Example of using the `solution` environment, with `answers` set

```
\question[5] Draw the function  $3x^2+4x+5$ !

\begin{solutionorgrid}[8cm]
\begin{tikzpicture}[baseline]
\begin{axis}[axis y line=center,axis x line=middle,
grid=both,xmax=5,xmin=-5,ymin=0,ymax=10,
xlabel=$x$,ylabel=$y$
xtick={-5,...,5},ytick={0,...,11},anchor=center]
\addplot[smooth,blue,thick,samples=100]
{3*x^2+4*x+5} ;
\end{axis}
\end{tikzpicture}
\end{solutionorgrid}
```

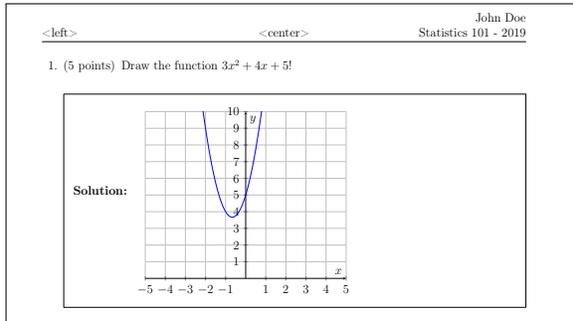


Figure 9: Example of using the `solutionorgrid` environment, with `answers` set

```
\gradetable[v][questions] vertically per question
\gradetable[h][questions] horizontally per question
\gradetable[v][pages] vertically per page
\gradetable[h][pages] horizontally per page
```

1. (2 points) What's the specific weight of air?  
 2. (2 points) What's the specific weight of air?

Question:	1	2	Total
Points:	2	2	4
Score:			

Figure 10: A grading table made with `\gradetable[h]` (requires two  $\LaTeX$  runs)

We will first work on the  $\LaTeX$  part before we automate the whole process. First we define a simple math question, as in Listing 2.

```
\begin{questions}
\question[5] Calculate!

\begin{parts}
\part[1] \((12345 + 67890 = \) \fillin[80235]
\end{parts}
```

Listing 2: A simple math exercise

We then use the `\qrcline` command from the `qrcline` package. This command takes just one parameter, the text to be encoded. In our case, this will be the numeric result of the calculation. For the vertical

```

\begin{questions}
\question[5] Calculate!

\begin{parts}
\part[1] \((12345 + 67890 = \) \fillin[80235]
\hfill\qrcode{80235}\vspace{2em}
\part[1] \((12345 + 67890 = \) \fillin[80235]
\hfill\qrcode{80235}\vspace{2em}
% ...

```

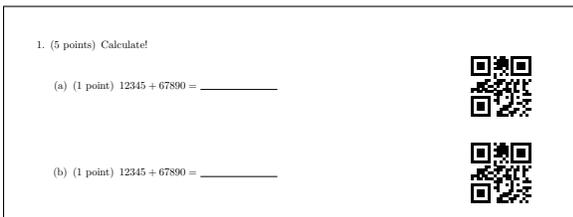


Figure 11: Adding and aligning QR codes

and horizontal alignment we use the `\hfill` and `\vspace` commands; see Figure 11 for sample input and output.

Next we develop the required Python code; see Listing 3. We create a function named `gen_exercise` to find two random integers and compute their sum, and return the  $\text{\LaTeX}$  string to typeset the exercise with the QR code. The code in the `\pyc` command parameter is only executed; it does not generate any printed text. The same holds for the `pycode` environment.

```

\pyc{from random import randrange}
\begin{pycode}
def gen_exercise():
    a = randrange(1000, 10000, 1)
    b = randrange(1000, 10000, 1)
    c = a + b
    a = str(a)
    b = str(b)
    c = str(c)
    return '\\(' + a + ' + ' + b
        + ' = \\) \\fillin[' + c
        + '] \\hfill\\qrcode{' + c
        + '}'\\vspace*{3em}'
\end{pycode}

```

Listing 3: Python (inside  $\text{\LaTeX}$ ) code to create an exercise

To automate our  $\text{\LaTeX}$  document with the Python code we use the `pythontex` package by Geoffrey Poore [2], which we presented in another talk at TUG 2019 (pp. 126–128 in this proceedings), and write the  $\text{\LaTeX}$  document shown (partially) in Listing 4.

```

\begin{questions}
\question[5] Calculate!

\begin{parts}
\part[1] \py{gen_exercise()}
\part[1] \py{gen_exercise()}
\part[1] \py{gen_exercise()}
\part[1] \py{gen_exercise()}
\end{parts}

```

Listing 4: Excerpt of the  $\text{\LaTeX}$  document using the `pythontex` code

In the  $\text{\LaTeX}$  `exam` document, inside the `parts` environment we use `\py` to call our `gen_exercise` function. It creates and returns the  $\text{\LaTeX}$  code desired. With the sequence `pdflatex`, `pythontex`, `pdflatex` we can then compile the final document. The output is similar to the previous one shown in Figure 11, but with randomly-generated numbers.

### 3 Summary

In this article we have presented the most important features of the `exam` class and shown how exams can be typeset with  $\text{\LaTeX}$  in a straightforward way. We have also shown how individual exercises can be created to allow more variability in the numerical values used in the exam.

Accompanying this article is the more extensive presentation held at TUG 2019 for which the interested reader is directed to the slides at [www.uweziegenhagen.de](http://www.uweziegenhagen.de).

### References

- [1] P. Hirschhorn. *Using the exam document class*, 2017. [ctan.org/pkg/exam](http://ctan.org/pkg/exam)
- [2] G. M. Poore. Python $\text{\TeX}$ : Reproducible documents with  $\text{\LaTeX}$ , Python, and more. *Comput. Sci. Disc.* 8(1), 2015. [ctan.org/pkg/pythontex](http://ctan.org/pkg/pythontex)

◇ Uwe Ziegenhagen  
Escher Str. 221  
50739 Cologne, Germany  
[ziegenhagen \(at\) gmail dot com](mailto:ziegenhagen@gmail.com)  
<https://www.uweziegenhagen.de>