The exesheet class and package

Antoine Missier
antoine.missier@ac-toulouse.fr

July 22, 2020, v1.0

Contents

1 Introduction

The exesheet *package* is used for typesetting exercise or exam sheets. In addition, the exesheet *class* loads the schooldocs package. This one makes adjustments for margins and title and defines various layout styles with particular header and footer, appropriate for exercise sheets (among others). See the schooldocs documentation for more details. The exesheet *class* is based on article and has the same options.

Many other packages are dedicated to exercise sheets. Most propose to encapsulate each exercise in an environment while exesheet begins each exercise with the `\exercise` command, which works like a subsection (with same features) and is suitable for documents consisting exclusively of exercises. The package provides also alternative formatting, more relevant for short exercises.

Another specificity of the exesheet package is specific settings for enumeration lists, different from LaTeX standard settings, usefull for the numbering of questions or answers inside an exercise.

Other packages provide often more or less elaborate mechanisms to manage the placement of answers. exesheet has no such ambitions: for all exercises of the sheet, we can display subject only, answer anly or both, but always at the place they are inserted in the source file. On the other hand this choice may be very flexible: we can do a correct version for all exercises together, or a correct version per exercise, per part (subpart of exercise), per question, per sub-question.

Finally the original fonctionality of this package is the display of a detailed marking scheme, with optional explanations, in the margin of exercise answers.

There are two possible options: `notoc` (see 2.4) and `nosetlist` (see 3.1).

In the current document, a frame is used to highlight examples beside the rest of the documentation.

2 Titles

2.1 The `\exercise` command

`\exercise` Each exercise begins with the `\exercise[`⟨*opt*⟩`]` command. This command typesets **Exercise**, as a document subsection, followed by automatic numbering, unique for the whole document. The optional parameter ⟨*opt*⟩ is used to put additional text on the same title line, for example to precise a subject or a marking scheme. `\exercise[ (to begin)]` yields:

Exercise 1 (to begin)

Try to use this first command now, it's easy.

To bring additional text closer to the exercise number, we can use the `\unskip` command which eliminates preceding space, see the following example, obtained with `\exercise[\unskip : calculation]`:

Exercise 2: calculation

Calculate $1 + 1$.

`\exercisename` The word *Exercise* has automatic translation in a few languages[1] when the `babel` package is loaded. It can be redefined, with `\renewcommand`, which should be placed after `\begin{document}`, or else we can use macros from the `babel` package (allowing dynamic language change), e.g. `\addto\captionssweedish{\def\exercisename{\"Ovning}}` (in the preamble).

`\labelexercise` This command calls `\exercisename` following by the exercise number. It can be redefined. For example, to add a period after the exercise number: `\renewcommand{\labelexercise}{\exercisename~\theexercise.}`

`\theexercise` To change only the numbering type, redefine the `\theexercise` command, based on the `exercise` counter.

`\labelexercisestyle` This macro (which is empty by default) allows to define a particular style for exercise titles. In the present document, we defined in the preamble: `\renewcommand{\labelexercisestyle}{\rmfamily\color{black}}`[2].

`\exercise*` The starred version `\exercise*[`⟨*opt*⟩`]{`⟨*label*⟩`}` allows to choose another ⟨*label*⟩ for a particular exercise and removes the numbering. For instance: `\exercise*[(Fermat theorem)]{Problem}` yields:

Problem (Fermat theorem)

Proove that there are no positive integers x, y, z such that $x^n + y^n = z^n$ for any integer n greater than 2.

2.2 The `\subpart` command

`\subpart` An exercise may contain several parts that we obtain with the `\subpart[`⟨*opt*⟩`]` command, typeset like a subsubsection.

Exercise 3

Part A (preliminary)

First of all, prepare your cup of tea.

Part B

Now you are ready to make the current exercise.

`\thesubpart` By default, the subpart numbering uses letters : A, B, C, etc. This numbering type can be redefined with the `\thesubpart` command based on the `subpart` counter, for instance `\renewcommand\thesubpart{\arabic{subpart}}`.

`\subpartname` `\labelsubpart` `\labelsubpartstyle` As for `\exercise`, the `\subpart` command uses `\subpartname` (with automatic translation in a few languages if `babel` is loaded), `\labelsubpart` and `\labelsubpartstyle`, which are editable.

[1] Translation is currently integrated into the package for the following languages: French, German, Spanish, Italian, Portuguese.

[2] In the present document, to highlight real sections and subsections titles, their color and font have been modified with the `\allsectionsfont` macro from the `sectsty` package.

`\subpart*` Like `\exercise*`, the starred version `\subpart*[`⟨*opt*⟩`]{`⟨*label*⟩`}` allows to freely typeset the subpart `{`⟨*label*⟩`}`, for instance `\subpart*{First part}`.

2.3 The `\annex` command

`\annex` The `\annex[`⟨*opt*⟩`]` command typesets the title **ANNEX**, in uppercase letters, centred and in the subsection style, with an optional parameter, added on the same line.

ANNEX (to return)

`\annexname` The word *Annex* has automatic translation in a few languages if babel package is loaded. It can be extended to other languages or modified by redifining `\annexname`.

`\annexstyle` The annex title style is set by the `\annexstyle` macro, defined as follows: `\newcommand\annexstyle{\MakeUppercase}`. This command may be redefined as one wants.

2.4 Exercise titles in table of contents (notoc package option)

`notoc` By default, the titles *Exercise*, *Part* or *Annex*, appear in the table of contents (or in the pdf file summary when hyperref package is used). To avoid this, you can set the boolean `notoc` to `true` or call the exesheet package (or class) with `notoc` option. But notice that title optional arguments will always be ignored in the table of contents.

2.5 Short exercises: the `\exe` command

`\exe` The `\exe` command starts an exercise by the abbreviation **Ex.** followed by the exercise number, without using sectioning commands, and the exercise body begins on the same line. An exercise starts a new paragraph without indentation.

Ex. 4 — This is a short exercise who can contain several paragraphs or questions however.

Here for example starts a new paragraph.

Ex. 5 — This is another short exercise.

`\exname` `\exlabel` `\exsepmark` The abbreviation *Ex* may be changed by redefining `\exname`. The `\exlabel` macro calls `\exname` following by a period then the exercise number, and `\exsepmark` typesets a long dash. We may change these features by redefining these commands.

`\exe*` The starred version prints no separator as shown below:

Ex. 6 Another short exercise without separator.

2.6 The `\points` command

`\points`

The `\points{⟨pts⟩}` command displays the number of points awarded to an exercise. It is intended to be entered in the optional argument of the `\exercise` command[3]. `\exercise[\points{5}]` yields:

> **Exercise 7** 5 points
>
> Try to read this whole document without drinking tea and you get five points.

`\pointsname` `\pointname` `\pointsstyle` `pointscolor`

Like before, the words *points* and in the singular *point* (if ⟨*pts*⟩ is inferior to 2) get automatic translation (with babel) in a few languages and are modifiables.

The style used by the `\points` command may be redefined with `\pointsstyle`. The color is set by `pointscolor` with the `\definecolor` command (from the xcolor package by Uwe Kern, loaded by exesheet), for instance:
`\definecolor{pointscolor}{named}{blue}`.

3 Enumerations and lists

3.1 List settings (nosetlist package option)

Enumeration lists are intended to represent questions and sub-questions inside exercises. For a good highlight, labels are typeset in bold. Moreover, they are left aligned, at the start of the line, without indentation, and the vertical space between items is increased compared with LaTeX standard lists. These settings are done by the `\setlist` command[4] from the enumitem package by Javier Bezos. Lists with `itemize` environment are kept in their default configuration[5].

> **Exercise 8**
>
> **1.** First question
>
> **(a)** First sub-question
>
> **(b)** Second sub-question
>
> **2.** Second question

`nosetlist`

We can avoid enumeration list modifications and restore the LaTeX default settings by passing the option `nosetlist` to the `exesheet` class or package.

3.2 List of exercises : the `exenumerate` environment

`exenumerate`

When an exercise sheet is made of short independant exercises, it would be ill-advised to display the complete title *Exercise* for each. In addition to the `\exe` command, previously presented, we provide an even lighter solution with the

[3] This command which has to be used in the optional argument of `\exercise`, as several others presented below, is incompatible with the memoir class which redefines section commands.

[4] Labels may also be changed by the `\labelenumi` and `\labelenumii` commands.

[5] The `french` otion of the babel package alters `itemize` lists behavior and uses long dash as labels for each list level. This behavior is problematic when mathematics follows the dash symbol because the latter may be confused wih the minus sign. Modifications of the `french` option (which supersedes `frenchb` of babel) are canceled to restore LaTeX default settings.

`exenumerate` environment, which is only an enumeration list in which spaces between items are further increased compared with those of `enumerate`. Here is an example (the main list is an `exenumerate` environment but sublists are produced with regular `enumerate` environments):

1. Translate the following sentences in english:

  (a) Nam dui ligula, fringilla a, euismod sodales, sollicitudin vel, wisi.

  (b) Nam lacus libero, pretium at, lobortis vitae, ultricies et, tellus.

2. Translate the following sentence in german:

  Donec aliquet, tortor sed accumsan bibendum, erat ligula aliquet magna, vitae ornare odio metus a mi.

3. Translate the following sentences in french:

  (a) Cum sociis natoque penatibus et magnis dis parturient montes, nascetur ridiculus mus.

  (b) Nulla ullamcorper vestibulum turpis.

The environment takes an optional parameter, like `enumerate`, which enables, among others, to typeset alternative list labels, e.g. `\begin{exenumerate}[A.]`. There are many other options (see the `enumitem` package documentation).

3.3 Items aligned by lines: `tablenum`

`tablenum` `tablenuma` `tablitem`

These three environments are used to typeset short questions (`tablenum`), subquestions (`tablenuma`) or `itemize` lists (`tablitem`) on the same line. They have the same syntax: `\begin{tablenum}[`⟨*opt*⟩`](`⟨*cols*⟩`)`. The ⟨*cols*⟩ parameter is the number of columns used by the environment. It must be *in parentheses*. This parameter can be omitted, then its value is 2. As for classic lists, each item begins with the `\item` command.

These three environments are defined by the `\NewTasks` macro, from the `tasks` package by Clemens Niederberger. They take an optional argument ⟨*opt*⟩ explained in the documentation of this package, e.g. `counter-format=tsk[r])` gives a roman numbering following by a closing parenthesis. There are also many possibilities to place items in an original way, for instance, the `\item*` command allows to specify the number of columns the item is supposed to span. In the following example, the five `\item` are placed in order between `\begin{tablenum}(3)` and `\end{tablenum}`. Notice that numbering is made line by line.

Exercise 9

Give the derivative of the following functions:

1. $f(x) = \dfrac{1-x^2}{\mathrm{e}^x + \mathrm{e}^{-x}}$, **2.** $g(x) = \ln\left(\dfrac{1-x}{1+x^2}\right)$, **3.** $h(x) = \displaystyle\int_0^1 \mathrm{e}^{xy}\,\mathrm{d}y$,

4. $k(x) = \displaystyle\sum_{i=1}^{\infty} \frac{1}{x^i}$, **5.** $l(x) = \displaystyle\int_{\frac{1}{x}}^{x} \frac{1}{\ln t}\,\mathrm{d}t$.

For `tablenuma`, labels are letters (a, b, c, ...) surrounded by parentheses. This cannot be modified globally (except by redefining the environment with `\RenewTasks`). If the exesheet package is called with option `nosetlist`, labels of `tablenum` and `tablenuma` environments are displayed in normal font with indentation, as for `enumerate`.

3.4 Items aligned by columns: `colsenum`

`colsenum`

To get numbering of items by columns, we have the `colsenum` environment: `\begin{colsenum}[`⟨*opt*⟩`]{`⟨*cols*⟩`}`. The mandatory parameter is the number of columns and the optional one will be passed to `enumerate`, allowing, for example, to change the type of numbering (a, A, etc.). To use this environment, we have to load the multicol package in the preamble.

Exercise 10

Give the derivative of the following functions:

1. $f(x) = \dfrac{1-x^2}{\mathrm{e}^x + \mathrm{e}^{-x}}$,

2. $g(x) = \ln\left(\dfrac{1-x}{1+x^2}\right)$,

3. $h(x) = \displaystyle\int_0^1 \mathrm{e}^{xy}\,\mathrm{d}y$,

4. $k(x) = \displaystyle\sum_{i=1}^{\infty} \frac{1}{x^i}$,

5. $l(x) = \displaystyle\int_{\frac{1}{x}}^{x} \frac{1}{\ln t}\,\mathrm{d}t$.

`colsenum*`

We will notice that, on each line, items are not necessarily well aligned, what can produce inelegant effects. On the other hand, the `colsenum` environment doesn't try to align columns from the bottom by adjusting space between items. If we want to get this (which is the default option in multicol) we have the `colsenum*` environment (same syntax than `colsenum`). Here what we get in that case, with the same exercise.

Exercise 11

Give the derivative of the following functions:

1. $f(x) = \dfrac{1-x^2}{\mathrm{e}^x + \mathrm{e}^{-x}}$,

2. $g(x) = \ln\left(\dfrac{1-x}{1+x^2}\right)$,

3. $h(x) = \displaystyle\int_0^1 \mathrm{e}^{xy}\,\mathrm{d}y$,

4. $k(x) = \displaystyle\sum_{i=1}^{\infty} \frac{1}{x^i}$,

5. $l(x) = \displaystyle\int_{\frac{1}{x}}^{x} \frac{1}{\ln t}\,\mathrm{d}t$.

We can see that these alignments are less good than those obtained by numbering of items by line. The numbering of items by columns may nevertheless be preferable when there are many items with variable heights, and a number of items which can be different from one column to the other. Moreover, an advantage of `colsenum` is that the choice of labels is automatic depending on the list level (and the language), unlike `tablenum` or `tablenuma`.

`colsitem`
`colsitem*`

For `itemize` lists, the environment `colsitem` produces items aligned by columns rather than by lines as for `tablitem`: `\begin{colsitem}[`⟨*opt*⟩`]{`⟨*cols*⟩`}`. The optional parameter, which is passed to the underlying `itemize` environment, allows to change the item label (bullet by default). And, as for `colsenum*`, the `colsitem*` environment produces an alignment of columns from the bottom.

4 Questions and solutions

4.1 The `questions` and `answers` environments

`questions`
`answers`

The package provides the two environments `questions` and `answers` to make appear or disappear questions and answers of exercises. Display is controlled by two booleans: `questions` and `answers`. Their default value is `true`.

`\questionsonly`
`\answersonly`

The `\questionsonly` command allows to display the questions without the answers and `\answersonly` displays the solutions without the questions[6].

`\correctionstyle`
`correctioncolor`
`\correctionname`

In the case (by default) where questions and answers are displayed both, answers are then typeset in the style `\correctionstyle`, which uses the color `correctioncolor`. This color may be choosen with the `\definecolor` macro (by default `\definecolor{correctioncolor}{rgb}{0,0.2,0.6}` = kind of dark blue). Moreover, in that case, the title **Correction** is displayed at the beginning of the `answers` environment. It is defined by the `\correctionname` macro (with translation in a few languages and that we can change:[7] for example we may prefer *Solution* than *Correction*). The style defined by `\correctionstyle` will apply to the title but also to the whole environment. Here an example.

Exercise 12

3 points

1. Is the exesheet package useful ?

2. Ins't there any other packages that deal with exercises ?

Correction

1. Yes, the exesheet package is very useful.

2. There are many other packages that deal with exercices, and give the ability to produce separatly questions and solutions, for example exercise by Paul Pichaureau, exercises by Roger Jud, exsheets (superseded by xsim) by Clemens Niederberger, exframe by Niklas Beisert, exam by Philip Hirschhorn, answers by Mike Piff and Joseph Wright, probsoln by Nicola Talbot, etc.

`\points`

When only answers are displayed, the text color remains black, the word *Correction* is not displayed and the `\points` macro is patched to suppress the display of points. An extended solution for a variable display of points will be disussed in section 5.2 with the `\totalpoints` macro.

[6] Of course we can also assign the booleans `questions` and `answers` directly with the LaTeX command `\setboolean`.

[7] Except for English, if you want to change package keywords you have to do it *after* `\begin{document}` because babel will modify these definitions at the end of the preamble.

4.2 More about `answers` environment

Internally, we have used the `\comment` and `\endcomment` macros from the versions package by Uwe Lück. Other excellent packages allow to manage selectively piece of code. Let us mention verbatim by Rainer Schöpf, comment by Victor Eijkhout, version by Donald Arseneau and Stephen Bellantoni, optional by Donald Arseneau and codesection by Matthias Pospiech.

The versions package used provides furthermore the `\includeversion{`⟨*env*⟩`}` and `\excludeversion{`⟨*env*⟩`}` macros who allow to make appear or disappear any environment ⟨*env*⟩ and these optional environments may be nested[8].

However the `questions` and `answers` environments perform another task, not only making appear or disappear piece of text. Indeed, there is a problem with the title *Correction*. In which format to typeset it and at which level to put it in the table of contents (or in the summary of the pdf file)? In fact it depends on which level the environment has been nested. We can make a single `answers` environment for all the exercises of the sheet or an `answers` environment for each exercise, for each exercise part, for each question or sub-question. In fact, the typeset of the title *Correction* and his level in the table of contents will be adjusted automatically.

`[`⟨*level*⟩`]` Nevertheless, we can imagine twisted situations in which the title level will not be correctly calculated. It is then possible to force the level of the title *Correction* with `\begin{answers}[`⟨*level*⟩`]`. The optional ⟨*level*⟩ parameter is defined as follows: 1 for section level titles, 2 for subsections (like *Exercise*), 3 for subsubsections (like *Part*), another number for lower levels (then they will not appear in the table of contents).

`answers*` The starred version `answers*` allows to make the title **Correction** completely disappear.

4.3 The `\question` command

`\question` Instead of the `questions` and `answers` environments, we can also use the simple `\question{`⟨*ques*⟩`}{`⟨*ans*⟩`}` macro in which display of ⟨*ques*⟩ and ⟨*ans*⟩ arguments is controlled by the same previous commands `\questionsonly` and `\answersonly`. It can be more readable (in the source code) when questions and answers are short. In the case of displaying both questions and answers, the *Correction* title will appear on a new line, in bold, as for levels lower than subsubsection in the `answers` environment (and without entries in the table of contents). But this command doesn't work with `verbatim` inside.

5 Marginal notes for marking scheme

The exesheet package allows the display of a marking scheme with comments and explanations for answers.

[8] The codesection package also allows such nesting, including in the preamble, as well as the optional package, but the latter only manages short optional code.

5.1 The `\pts` command

`\pts` When exercises are typeset with the `\exe` macro, or as a list with the `exenumerate` environment, the marking scheme is displayed in the margin, on the line where we put the `\pts{`⟨*num*⟩`}` command (in general the first line of the exercise). The ⟨*num*⟩ parameter is the number of points assigned to the exercise.

(3 pts) **Ex. 13** — The first exercise with marking sheme.

(1.5 pt) **Ex. 14** — The second one.

`\ptsname` `\ptname` `ptscolor` `\ptsstyle` The abbreviation pts (or pt when the number of points is inferior to 2) is automatically added thanks to the macro `\ptsname` or `\ptname` (translated in a few languages if babel is loaded). The display color of the points is defined by `ptscolor` which can be changed with `\definecolor`: by default `\definecolor{ptscolor}{named}{red}`. The display style is defined by `\ptsstyle`.

`\displaypts` The display of the marking scheme, as above, is internally controlled by the `marginpts` boolean. The `\displaypts` command, which has to be placed in the preamble, defines `marginpts` to `true`, places marginal notes to the left (and not to the right which is the default behavior of LaTeX) and shifts the text body right to increase the left margin compared with the right one, with a ratio 3:2, as it is in the present document. This ratio is defined by the `geometry` macro from the geometry package (by Hideo Umeki) and is valid for the whole document.

5.2 The `\totalexe`, `\note` and `\totalpoints` commands

`\totalexe` `\note` To give a more detailed marking sheme, the exesheet package provides some extended commands: `\totalexe{`⟨*num*⟩`}` to display the total number of points of an exercise and especially `\note[`⟨*pts*⟩`]{`⟨*comment*⟩`}`, to indicate some marking detail in answers. The optional parameter ⟨*pts*⟩ is the number of points of the question and the mandatory parameter ⟨*comment*⟩ is used to give details about the marking scheme. Below the title has been obtained with `\exercise[\totalexe{4}]` and the first comment with
`\note[1]{0,5 for the anti-derivative\\0,5 for simplification of $\ln$}`.

4 pts

Exercise 15

For each following question, say if the assertion is true or false. Justify the answer carefully.

1. $\displaystyle\int_0^{\sqrt{3}} \frac{1}{x+\sqrt{3}}\,\mathrm{d}x = \ln 2,$ **2.** $\displaystyle\int_2^{\mathrm{e}} \frac{1}{x\ln x}\,\mathrm{d}x = -\ln 2,$

3. The function F defined on $\mathbf{R}$ by $F(x) = \displaystyle\int_0^x \frac{1}{,t^2+t+1}\,\mathrm{d}t$, is increasing on $\mathbf{R}$.

Correction

1
0,5 for the anti-derivative
0,5 for simplification of $\ln$

1. We calculate:

$$\int_0^{\sqrt{3}} \frac{1}{x+\sqrt{3}}\,\mathrm{d}x = \Big[\ln\left(x+\sqrt{3}\right)\Big]_0^{\sqrt{3}} = \ln\left(2\sqrt{3}\right) - \ln\sqrt{3} = \ln\left(\frac{2\sqrt{3}}{\sqrt{3}}\right) = \ln 2.$$

TRUE.

1.5
1 for the anti-derivative
0.5 for the integral value

we could also see that $\frac{1}{x\ln x} > 0$
on $[2, \mathrm{e}]$ while $-\ln 2 < 0$

2. We have $\dfrac{1}{x\ln x} = \dfrac{\frac{1}{x}}{\ln x} = \dfrac{u'(x)}{u(x)}$ with $u(x) = \ln x$, which is positive on $[2, \mathrm{e}]$. Hence

$$\int_2^{\mathrm{e}} \frac{1}{x\ln x}\,\mathrm{d}x = \Big[\ln(\ln x)\Big]_2^{\mathrm{e}} = \ln(\ln \mathrm{e}) - \ln(\ln 2) = \ln 1 - \ln(\ln 2) = -\ln(\ln 2).$$

FALSE.

1.5

3. The function F defined on $\mathbf{R}$ by

$$F(x) = \int_0^x \frac{1}{t^2+t+1}\,\mathrm{d}t$$

0.5 for F'
1 for the sign of F' and conclusion

is derivable on $\mathbf{R}$ and its derivative is such that $F'(x) = \frac{1}{x^2+x+1}$. The denominator is a quadratic polynomial, always positive because its discriminant is $\Delta = -3 < 0$. Thus F is increasing on $\mathbf{R}$.
TRUE.

In the comment of answer 2, a wider vertical space is produced at line break with the optional argument `\\[2ex]`[9]. The last comment, not placed beside the number of points of answer 3, has been produced by placing on the first line after the formula: `\note{0.5 for $F'$\\1 for the sign of $F'$ and conclusion}`.

`\note*` If we don't want to add comment, we have the `\note*{`⟨*pts*⟩`}` command (better thant `\note[`⟨*pts*⟩`]{}`), used here in answer 3: `\note*{1.5}`.

`markingcolor` `\markingstyle` `\ptsboxlength` The display color of the points, in `\totalexe` and `\note`, is defined by `markingcolor` and the style by `\markingstyle`, which are modifiables. The oval box produced by `\totalexe` is obtained with the `\ovalbox` command of the `fancybox` package by Timothy Van Zandt, with corner arcs set by `\cornersize{1}`. The box length is not adjusted to content but depends of the value of `\ptsboxlength` in order to keep a uniform appearance from one exercise to the other.

`notecolor` `\notestyle` `\noteragged` Comment notes are typeset by default in a dark green color defined by `\definecolor{notecolor}{rgb}{0.0,0.4,0.0}`. The style of the comment is set by the `\notestyle` macro and itself uses the `\noteragged` macro. The latter is set by default as `\raggedleft` to obtain a *right* alignment (what is customary for notes in the left margin).

`\displaypoints` Display of points and comments is optional. It is internally controlled by the booleans `marginpoints` and `marginfullnotes`. The `\displaypoints` macro, to place in the preamble, sets `marginpoints` to `true`, chooses the left margin for

[9] When a marginal note at the end of a page, is followed by another one at the top of the next page, the latter is shifted down. We can raise this note, by adding at the line break a negative vertical space : `\note*{\\[-5ex] 2}`.

display marginal notes and sets the ratio between left and right margins to 3:2[10]. Points will then be displayed without the comments.

\displaynotes The \displaynotes[⟨*align*⟩] macro, to be placed also in the preamble, defines both booleans to true and allows then the complete display of points and comments. The ratio between left/right margins is then set to 5/1[11]. The optional parameter ⟨*align*⟩ is used to define alignment: \raggedleft by default, but we could use \centering or \raggedright (not recommended in left margin).

\displaynotesright The \displaynotesright[⟨*align*⟩] macro displays a detailed marking scheme with comments in the right margin, with the same type of settings as \displaynotes but an optional parameter ⟨*align*⟩ set by default to \raggedleft.

\totalpoints The \totalpoints{⟨*num*⟩} macro is intended to replace \points when using a detailed marking scheme. When the scale is not displayed (marginpoints boolean to false), it uses \points and when the scale is displayed, it uses \totalexe. For example, in the exercise 15, we should use \totalpoints rather than \totalexe, because, when the detailed marking scheme is not displayed, the total points will be typeset as in the exercise 12 rather than in the margin.

6 Implementation

6.1 Class and package basic instructions

The exesheet class is based on the article class and gives it all its options, except explicit options notoc and nosetlist.

```
⟨*class⟩
\DeclareOption{notoc}{\PassOptionsToPackage{notoc}{exesheet}}
\DeclareOption{nosetlist}{\PassOptionsToPackage{nosetlist}{exesheet}}
\DeclareOption*{\PassOptionsToClass{\CurrentOption}{article}}
\ProcessOptions \relax
\LoadClass{article}
\RequirePackage{exesheet}
\RequirePackage{schooldocs}
⟨/class⟩
```

Then we define initialization instructions of the exesheet package.

The shortlabel option in the enumitem package allows to use labels like in the enumerate package e.g. 1., a), A. etc.

```
⟨*package⟩
\RequirePackage{ifthen}
\newboolean{notoc}
\newboolean{nosetlist}
\DeclareOption{notoc}{\setboolean{notoc}{true}}
\DeclareOption{nosetlist}{\setboolean{nosetlist}{true}}
\ProcessOptions \relax
\RequirePackage{xcolor}
\RequirePackage[shortlabels]{enumitem}
```

[10] So that the effect on the margin ratio is correct, the \displaypoints macro must be placed after other commands that could alter the page geometry, for example the \pagestyle of the schooldocs package.

[11] The present document kept a ratio of 3/2 because the ltxdoc class, used here, leaves large enough margins and previous examples are not concerned with display of comments.

```
\RequirePackage{tasks}
\RequirePackage{versions}
%\@ifpackageloaded{doc}{}{\RequirePackage{verbatim}}
\RequirePackage{geometry}
\RequirePackage{fancybox}

```

6.2 Internationalization

Here we define the keywords and their translation in French, German, Spanish Italian, Portuguese, thanks to the `\addto\captions`⟨*language*⟩ macro of the babel package.

Translations are placed in a macro `\exetranslate` because, if exesheet is called before babel (this is necessarily the case for the exesheet class), then these translations should only happen after having loaded the babel package, therefore the `\AtBeginDocument`. On the other hand, if the package is loaded *after* babel, `AtBeginDocument{\exetranslate}` doesn't work, curiously. But calling immediately `\exetranslate` works in that case.

Accented characters cannot be used here because they are not recognized if inputenc is loaded after exesheet, despite the `\AtBeginDocument`. So we have used the basic LaTeX commands to produce them.

TODO: give the choice to use polyglossia instead of babel; see the translations or multilang packages.

```
\def\exercisename{Exercise}
\def\subpartname{Part}
\def\annexname{Annex}
\def\exname{Ex}
\def\pointsname{points}
\def\pointname{point}
\def\correctionname{Correction}
\def\ptsname{pts}
\def\ptname{pt}

\newcommand\exetranslate{%
    \@ifpackageloaded{babel}{%
        \addto\captionsfrench{%
            \def\exercisename{Exercice}
            \def\subpartname{Partie}
            \def\annexname{Annexe}
            \def\exname{Ex}
            \def\pointsname{points}
            \def\pointname{point}
            \def\correctionname{Correction}
            \def\ptsname{pts}
            \def\ptname{pt}
            }
        \addto\captionsgerman{%
            \def\exercisename{\"Ubung}
            \def\subpartname{Teil}
            \def\annexname{Anhang}
            \def\exname{\"Ub}
            \def\pointsname{Punkte}
            \def\pointname{Punkt}
            \def\correctionname{Verbesserung}
            \def\ptsname{Pkte}
            \def\ptname{Pkt}
            }
        \addto\captionsspanish{%
            \def\exercisename{Ejercicio}
            \def\subpartname{Parte}
            \def\annexname{Anexo}
            \def\exname{Ej}
            \def\pointsname{puntos}
            \def\pointname{punto}
            \def\correctionname{Correcci\'on}
            \def\ptsname{ptos}
            \def\ptname{pot}
            }
        \addto\captionsitalian{%
            \def\exercisename{Esercizio}
            \def\subpartname{Parte}
            \def\annexname{Ansesso}
            \def\exname{Es}
            \def\pointsname{punti}
            \def\pointname{punto}
            \def\correctionname{Correzione}
            \def\ptsname{pti}
            \def\ptname{pt}
            }
        \addto\captionsportuges{%
            \def\exercisename{Exerc\'icio}
            \def\subpartname{Parte}
            \def\annexname{Anexo}
            \def\exname{Ex}
            \def\pointsname{pontos}
            \def\pointname{ponto}
            \def\correctionname{Corre\c c\~ao}
            \def\ptsname{pts}
            \def\ptname{pt}
            }
        }{}
    }

\AtBeginDocument{\exetranslate} % if loaded before babel
\exetranslate % necessary when loaded after babel

```

6.3 Colors

The color `pointscolor` is used by the macro `\points` and `markingcolor` by the macros `\totalexe`, `\note*` and also for the optional argument of `\note`; `notecolor` is used by the mandatory argument of `\note`.

```
\definecolor{pointscolor}{named}{red}
\definecolor{ptscolor}{named}{red}
\definecolor{markingcolor}{named}{red}
```

```
\definecolor{notecolor}{rgb}{0.0, 0.4, 0.0} % kind of dark green
\definecolor{correctioncolor}{rgb}{0,0.2,0.6} % kind of dark blue

```

6.4 Titles

The `exercise` counter numbers exercises for the whole document regardless of any section. To reset the counter at some point, just write `\setcounter{exercise}{0}` and if we want an automatic reset at each section, add in the preamble `\makeatletter \@addtoreset{exercise}{section} \makeatother`.

The parts counter depends on `exercise` and is reset at each new exercise.

The `\labelexercisestyle` and `\labelsubpartstyle` commands are empty, but allow to customize the style, for instance: `\renewcommand\labelexercisestyle{\sffamily}`.

By default, the table of contents displays the titles of exercises and parts. The `notoc` boolean is used to modify this behavior. If we want to display only exercise titles but not parts, we can place in the preamble `\setcounter{tocdepth}{2}`.

`\exercise`

```
\newcounter{exercise}

\newcommand{\labelexercise}{\exercisename~\theexercise}
\newcommand{\labelexercisestyle}{}
\newcommand*{\@exercise}[1][]{%
    \refstepcounter{exercise}
    \subsection*{\labelexercisestyle\labelexercise\ #1}
    \ifthenelse{\boolean{notoc}}{}{
        \addcontentsline{toc}{subsection}{\labelexercise}
        }
    }
\newcommand*{\@@exercise}[2][]{%
    \subsection*{\labelexercisestyle #2 #1}
    \setcounter{subpart}{0} % resets the parts counter
    \ifthenelse{\boolean{notoc}}{}{
        \addcontentsline{toc}{subsection}{#2}
        }
    }
\newcommand{\exercise}{\@ifstar{\@@exercise}{\@exercise}}

```

`\subpart`

```
\newcounter{subpart}[exercise] %
\renewcommand{\thesubpart}{\Alph{subpart}}

\newcommand{\labelsubpart}{\subpartname~\thesubpart}
\newcommand{\labelsubpartstyle}{}
\newcommand*{\@subpart}[1][]{%
    \refstepcounter{subpart}%
    \subsubsection*{\labelsubpartstyle\labelsubpart\ #1}
    \ifthenelse{\boolean{notoc}}{}{
        \addcontentsline{toc}{subsubsection}{\labelsubpart}
        }
    }
\newcommand*{\@@subpart}[2][]{%
    \subsubsection*{\labelsubpartstyle #2 #1}
    \ifthenelse{\boolean{notoc}}{}{
        \addcontentsline{toc}{subsubsection}{#2}
        }
    }
\newcommand{\subpart}{\@ifstar{\@@subpart}{\@subpart}}

```

\annex

```
\newcommand{\annexstyle}{\MakeUppercase}
\newcommand*{\annex}[1][]{%
    \subsection*{\mbox{}\hfill\annexstyle{\annexname} #1\hfill\mbox{}}
    \ifthenelse{\boolean{notoc}}{}{
        \addcontentsline{toc}{subsection}{\annexname}
        }
    }

```

\exe

```
\newcommand{\exlabel}{\exname.~\theexercise}
\newcommand{\exsepmark}{---}
\newcommand{\@exe}{\textbf{\exlabel~\exsepmark}~}
\newcommand{\@@exe}{\textbf{\exlabel}~}
\newcommand{\exe}{\bigskip\par\noindent\refstepcounter{exercise}
    \@ifstar{\@@exe}{\@exe}
    }

```

\points (ini)

```
\newcommand{\pointsstyle}{%
    \small\mdseries\sffamily\color{pointscolor}\fbox}
\newcommand*{\points}[1]{\hfill
    \pointsstyle{#1~%
        \ifthenelse{\lengthtest{#1 cm < 2cm}}{\pointname}{\pointsname}%
        }
    }

```

Percent symbols are necessary to avoid spaces between the `\fbox` and its inner text. Note that, without `\lengthtest`, the test `#1 < 2` doesn't work with decimal numbers.

6.5 Enumerations and lists

When using babel with the `french` option[12], `itemize` lists are modified with the same dash label for each list level. These modifications are cancelled here to restore default LaTeX `itemize` lists (labels and spaces). As for `\exetranslate`, we have create the `\standardfrenchlists` command who must be called into `AtBeginDocument` or not, depending on whether exesheet is loaded before babel or after.

[12] The `french` option of babel superseeds the `frenchb` option.

The `\setlist` command comes from the enumitem package (`\setenumerate` is deprecated). By default `itemsep=1ex` for lists of first level, and `leftmargin=1.5em` allows to align labels on the start of lines.

```
\newcommand\standardfrenchlists{%
    \@ifpackagewith{babel}{frenchb}{
        \frenchbsetup{StandardLists=true}}{}
    \@ifpackagewith{babel}{french}{
        \@ifundefined{frenchsetup}{
            \frenchbsetup{StandardLists=true}}{
            \frenchsetup{StandardLists=True}}
        }{}
    }
\ifthenelse{\boolean{nosetlist}}{}{
    \AtBeginDocument{% if loaded before babel package
        \standardfrenchlists}
    \standardfrenchlists % necessary when loaded after babel
    \setlist[enumerate]{font=\bfseries}
    \setlist[enumerate,1]{topsep=1.5ex plus 1ex minus 1ex,leftmargin=1.5em}
    }

```

`\exenumerate`

```
\newenvironment{exenumerate}[1][]{%
    \setlist[enumerate]{font=\bfseries}
    \setlist[enumerate,1]{leftmargin=1.5em,
        itemsep=3ex plus 1ex minus 1ex,topsep=3ex plus 1ex minus 1ex}
    \setlist[enumerate,3]{noitemsep,nolistsep}
    \setlist[itemize]{noitemsep,nolistsep}
    \begin{enumerate}[#1]
    }{\end{enumerate}}

```

`tablenum`
`tablenuma`

The `\NewTasks` command comes from the tasks package. It allows to define the environments `tablenum`, `tablenuma` and `tablitem`. The horizontal spaces are adjusted to get a good alignment with items of other `enumerate` (or `itemize`) environments.

```
\ifthenelse{\boolean{nosetlist}}{
    \NewTasks[counter-format=tsk[1].,
        column-sep=1em,
        after-item-skip=0.5ex plus 0.5ex minus 0.5ex]{tablenum}[\item](2)
    \NewTasks[counter-format=(tsk[a]),
        column-sep=1em,label-align=right,
        item-indent=2.15em,label-width=1.6em,label-offset=0.5em,
        after-item-skip=0.5ex plus 0.5ex minus 0.5ex]{tablenuma}[\item](2)
    }{% by default
    \NewTasks[counter-format=tsk[1].,label-format=\bfseries,
        column-sep=1em,label-align=right,
        item-indent=1.5em,label-width=1em,label-offset=0.5em,
        after-item-skip=0.5ex plus 0.5ex minus 0.5ex]{tablenum}[\item](2)
    \NewTasks[counter-format=(tsk[a]),label-format=\bfseries,
        column-sep=1em,label-align=right,
        item-indent=2.15em,label-width=1.6em,label-offset=0.5em,
        after-item-skip=0.5ex plus 0.5ex minus 0.5ex]{tablenuma}[\item](2)
    }

```

tablitem

```
\NewTasks[label=\labelitemi,
    label-align=right,
    item-indent=2.3333em,label-offset=0.5em,
    after-item-skip=0.5ex plus 0.5ex minus 0.5ex]{tablitem}[\item](2)

```

colsenum
colsenum*

```
\newenvironment{colsenum}[2][]{%
    \setlength{\multicolsep}{2ex}
    \raggedcolumns % default is \flushcolumns
    \begin{multicols}{#2} % #2 = number of columns
    \begin{enumerate}[#1] % #1 = options of enumerate
    }{
    \end{enumerate}
    \end{multicols}
}

\newenvironment{colsenum*}[2][]{%
    \setlength{\multicolsep}{2ex}
    \begin{multicols}{#2} % #2 = number of columns
    \begin{enumerate}[#1] % #1 = options of enumerate
    }{
    \end{enumerate}
    \end{multicols}
}

```

colsitem
colsitem*

```
\newenvironment{colsitem}[2][]{%
    \setlength{\multicolsep}{2ex}
    \raggedcolumns
    \begin{multicols}{#2}
    \begin{itemize}[#1]
    }{
    \end{itemize}
    \end{multicols}
}

\newenvironment{colsitem*}[2][]{%
    \setlength{\multicolsep}{2ex}
    \begin{multicols}{#2}
    \begin{itemize}[#1]
    }{
    \end{itemize}
    \end{multicols}
}

```

6.6 Questions and answers

\questionsonly
\answersonly

The questions and answers booleans control the display of corresponding environments. When created, a boolean has the `false` value by default, but we define them to `true`. The `\questionsonly` and `\answersonly` macros are used as user interface to display only one of the two environments.

```
\newboolean{questions}
\newboolean{answers}
\setboolean{questions}{true}
\setboolean{answers}{true}
\newcommand{\questionsonly}{
    \setboolean{questions}{true}\setboolean{answers}{false}}
\newcommand{\answersonly}{
    \setboolean{questions}{false}\setboolean{answers}{true}}

```

`questions` It is the `\comment` and `\endcomment` macros, provided by the versions package, that allow the magic of conditional displays (we can also find them in the verbatim or version packages). The noteworthy codesection package, allows to encapsulate optional code between the macros `\BeginCodeSection{`⟨*skip*⟩`}` and `\EndCodeSection{`⟨*skip*⟩`}`, both in the text body and in the preamble, but these macros cannot be used inside an environment as we did here for `\comment` and `\endcomment`.

```
\newcounter{exe@ini}
\newcounter{subpart@ini}

\newenvironment{questions}{
    \ifthenelse{\boolean{questions}}{
        \setcounter{exe@ini}{\value{exercise}}
        \setcounter{subpart@ini}{\value{subpart}}
        }{\comment}}%
    {\ifthenelse{\boolean{questions}}{}{\endcomment}}

```

`answers` The internal macro `\set@toclevel` calculates the title level of the word *Correction* to display at the start of an `answers` environment (when `questions` and `answers` are displayed together). The principle is to compare the state of the counters `exercise` and `subpart` with those saved at the time of the call of `questions`. The `\@enumdepth` counter indicates the `enumerate` list level in which we are (0 = out of lists). The optional parameter of the `answers` environment allows to force this title level.

```
\newcounter{@toclevel}
\newcommand{\set@toclevel}[1][]{
    \ifthenelse{\equal{#1}{}}{
        \ifthenelse{\value{exercise} > \value{exe@ini}}{
            \setcounter{@toclevel}{1}
        }{\ifthenelse{\equal{\the\@enumdepth}{0}}{
            % we're not in an enumerate environment
            \ifthenelse{\(\value{subpart} > \value{subpart@ini}\)
                \or \(\value{subpart} = 0\)}{
                \setcounter{@toclevel}{2}
            }{\setcounter{@toclevel}{3}}
          }{\setcounter{@toclevel}{4}}}
    }{\setcounter{@toclevel}{#1}}}

```

```
\newcommand{\correctionstyle}{\color{correctioncolor}}

\newenvironment{answers}[1][]{% #1 is the optional level
    \ifthenelse{\boolean{answers}}{%
        \ifthenelse{\boolean{questions}}{%
            \set@toclevel[#1]
            \ifthenelse{\value{@toclevel} = 1}{
                \section*{\correctionstyle\correctionname}
                \ifthenelse{\boolean{notoc}}{}{
                    \addcontentsline{toc}{section}{\correctionname}}
                \setcounter{exercise}{0}
            }{\ifthenelse{\value{@toclevel} = 2}{%
                  \subsection*{\correctionstyle\correctionname}
                  \ifthenelse{\boolean{notoc}}{}{
                      \addcontentsline{toc}{subsection}{\correctionname}}
                  \setcounter{subpart}{0}
              }{\ifthenelse{\value{@toclevel} = 3}{%
                    \subsubsection*{\correctionstyle\correctionname}
                    \ifthenelse{\boolean{notoc}}{}{
                        \addcontentsline{toc}{subsubsection}{
                            \correctionname}}
                    }{\par\textbf{\correctionstyle\correctionname}\par
                    }%
              }%
            }%
        \correctionstyle%
        }{}%
    }{\comment}
}{\ifthenelse{\boolean{answers}}{}{\endcomment}}

\newenvironment{answers*}{\ifthenelse{\boolean{answers}}{}{\comment}}%
    {\ifthenelse{\boolean{answers}}{}{\endcomment}}

```

In the answers environment, if we place \correctionstyle before \subsubsection, the preceding vertical space may be too wide.

\question

```
\newcommand{\question}[2]{%
    \ifthenelse{\boolean{questions}}{#1}{}
    \bgroup
    \ifthenelse{\boolean{answers}}{
        \ifthenelse{\boolean{questions}}{
            \ifx#2\empty\else
                \par\correctionstyle\textbf{\correctionname}\par
            \fi
            }{}
        #2}{}
    \egroup
    }

```

\points (patched)

```
\let\@oldpoints\points
\renewcommand*{\points}[1]{%
```

```
    \ifthenelse{\boolean{questions}}{\@oldpoints{#1}}{}}

```

6.7 Marginal notes

The commands \displaypts, \displaypoints and \displaynotes change the ratio between left and right margins[13].

\pts
\displaypts

```
\newboolean{marginpts}
\newcommand*{\pointmark}[1]{%
    \ifthenelse{\lengthtest{#1 cm < 2cm}}{#1 \ptname}{#1 \ptsname}}
\newcommand{\ptsstyle}[1]{%
    \footnotesize\centering\sffamily\color{ptscolor} (#1)}
\newcommand*{\pts}[1]{%
    \ifthenelse{\boolean{marginpts}}{%
        \mbox{}%
        \marginpar{\hspace{0pt}%
            \ptsstyle{\pointmark{#1}}}%
        }{}%
    \ignorespaces
    }
\newcommand{\displaypts}{%
    \reversemarginpar
    \geometry{hmarginratio=3:2}
    \setboolean{marginpts}{true}
    }

```

\totalexe

In the following macros using \marginpar, percent symbols and \ignorespaces are necessary to avoid too much space in the text (or the margin) where these macros are inserted.

```
\newlength{\ptsboxlength}
\setlength{\ptsboxlength}{3.1em}
\cornersize{1}
\newcommand*{\totalexe}[1]{%
    \ifthenelse{\boolean{marginpoints}}{%
        \mbox{}%
        \marginpar{\markingstyle{\ovalbox{%
            \makebox[\ptsboxlength]{\pointmark{#1}}}%
            }}%
        }{}%
    \ignorespaces
    }

```

\note

The booleans marginpoints and marginfullnotes control the display of marginal notes. If marginpoints is false, marginfullnotes will be ignored. The \noteragged command is initialized outside \displaynotes such that we can use the \note command without \displaynotes (see further).

```
\newboolean{marginpoints}
```

[13] So that the effect on the margin ratio is correct, these macros must be called, in the preamble, after other commands that also could alter the page geometry.

```
\newboolean{marginfullnotes}

\newcommand{\markingstyle}[1]{\hspace{0pt}\footnotesize\sffamily%
    \centering\color{markingcolor}\textbf{#1}}
\newcommand{\noteragged}{\raggedleft}
\newcommand{\notestyle}[1]{\hspace{0pt}\footnotesize\sffamily%
    \noteragged\noindent\color{notecolor} #1}
\newcommand{\@note}[2][]{%
    \ifthenelse{\boolean{marginpoints}}{%
        \mbox{}%
        \marginpar{%
            \ifthenelse{\equal{#1}{}}{}{\markingstyle{#1}\\}%
            \ifthenelse{\boolean{marginfullnotes}}{\notestyle #2}{}%
            }%
        }{}%
    \ignorespaces
    }
\newcommand{\@@note}[1]{%
    \ifthenelse{\boolean{marginpoints}}{%
        \mbox{}%
        \marginpar{\markingstyle{#1}}%
        }{}%
    \ignorespaces
    }
\newcommand{\note}{\@ifstar{\@@note}{\@note}}

```

\displaypoints
\displaynotes
\displaynotesright

In \displaynotes, the additional length 1 in matches the default free space to the left of \oddsidemargin.

```
\newcommand{\displaypoints}{%
    \reversemarginpar
    \geometry{hmarginratio=3:2}
    \setboolean{marginpoints}{true}
    }

\newcommand*{\displaynotes}[1][\raggedleft]{%
    \reversemarginpar
    \renewcommand{\noteragged}{#1}
    \geometry{hmarginratio=5:1}
    \setlength{\marginparwidth}{\oddsidemargin}
    \addtolength{\marginparwidth}{1in}
    \addtolength{\marginparwidth}{-\marginparsep}
    \setlength{\marginparwidth}{0.8\marginparwidth}
    \setboolean{marginpoints}{true}
    \setboolean{marginfullnotes}{true}
    }

\newcommand*{\displaynotesright}[1][\raggedright]{%
    \normalmarginpar
    \renewcommand{\noteragged}{#1}
    \geometry{hmarginratio=1:5}
    \setlength{\marginparwidth}{\paperwidth}
    \addtolength{\marginparwidth}{-\textwidth}
    \addtolength{\marginparwidth}{-\oddsidemargin}
    \addtolength{\marginparwidth}{-\marginparsep}
    \addtolength{\marginparwidth}{-1in}
    \setlength{\marginparwidth}{0.8\marginparwidth}
    \setboolean{marginpoints}{true}
    \setboolean{marginfullnotes}{true}
    }
```

The formatting of marginal notes can possibly be made more regular with the ragged2e package, not loaded by exesheet. We will then have to enter, as an optional parameter of `\displaypoints`: `\RaggedLeft`, `\Centering`, `\RaggedRight` or `justifying`.

`\totalpoints`

```
\newcommand{\totalpoints}{%
    \ifthenelse{\boolean{marginpoints}}{\totalexe}{\points}}
⟨/package⟩
```