The exesheet class and package
Antoine Missier
antoine.missier@ac-toulouse.fr
2023/12/30, v2.5

Contents

1 Introduction 2

2 Titles 2
  2.1 The \exercise command ................................. 2
  2.2 The \subpart command ................................ 3
  2.3 The \annex command ................................ 4
  2.4 Titles in the table of contents ..................... 4
  2.5 Short exercises: the \exe command .................. 4

3 Enumerations and lists 5
  3.1 List settings ........................................... 5
  3.2 List of exercises: the exenumrate environment .... 5
  3.3 Items aligned by row: tablenum1, tablenuma, tablitem .. 6
  3.4 Items aligned by column: colsenum, colsiem ........ 7

4 Questions and solutions 8
  4.1 Environments questions and answers ................ 8
  4.2 More about answers environments .................... 9
  4.3 Commands question, answer and answerspace ......... 10

5 Marking scheme commands 10
  5.1 The \points command .................................. 10
  5.2 The \pts command ..................................... 11
  5.3 Commands \totalexe, \note* and \note ................. 11
  5.4 Margin notes options ................................. 13
  5.5 The \totalpoints command ................................ 14
  5.6 Marking scheme consistency checking .............. 14

6 Options and comparison with other packages 15
  6.1 Summary of available options ....................... 15
  6.2 Alternative commands ................................ 15
  6.3 Comparison with other packages ..................... 16

7 Implementation 17
  7.1 Options and required packages ....................... 17
  7.2 Internationalization .................................. 19
  7.3 Titles .................................................. 21
  7.4 Enumerations and lists ............................... 23
  7.5 Questions and answers ................................ 25
  7.6 Marking scheme options processing ................. 29
  7.7 Marking scheme commands ......................... 35
1 Introduction

The \texttt{exesheet} package is designed for typesetting exercise or exam sheets. Additionally, the \texttt{exesheet} class loads the \texttt{schooldocs} package. The latter makes adjustments to margins and titles, and defines various layout styles with specific headers and footers suitable for exercise sheets, among other uses. Refer to the documentation of the \texttt{schooldocs} package for more details. The \texttt{exesheet} class is build upon the \texttt{article} class and forwards any unknown options to it.

There are many other packages dedicated to exercise sheets. In section 6.3 we provide an overview of some of their functionalities. Most of them suggest encapsulating each exercise within an environment. In contrast, \texttt{exesheet} starts each exercise with \texttt{\exercise}, which functions similarly to a subsection (with the same features) and is suitable for documents that primarily consist of exercises. The package also offers alternative ways to introduce exercises, which are more appropriate for shorter exercises.

Another distinctive feature of the \texttt{exesheet} package is its specific settings for enumeration lists, which are suitable for numbering questions or answers within an exercise.

For all exercises within the sheet, you can display only the questions, only the answers, or both, all while preserving their placement as they appear in the source file. This choice allows for great flexibility: you can create a correct version for all exercises collectively, or individual corrections per exercise, per part (subpart of exercise), per question, per sub-question.

The ability to hide questions or answers is found in many packages, but the main interest of \texttt{exesheet} is to be able to display or not a detailed scoring guide, along with correction instructions. This is very useful for grading papers with multiple graders. Furthermore \texttt{exesheet} can check the consistency of the scale.

Many settings can be customized, and various options are available to manage the output document. These options rely on the key-val mechanism: \texttt{key=value}. These options can be applied when calling the class or the package, e.g.

\begin{verbatim}
\documentclass[a4paper,11pt,output=answers,display=pts]{exesheet}
\end{verbatim}

or later using the command \texttt{\exesheetset{(options)}}. In the example above, \texttt{a4paper,11pt} are options that are passed to the underlying \texttt{article} class.

\textit{In the current document, a frame is utilized to emphasize output examples.}

2 Titles

2.1 The \texttt{\exercise} command

The \texttt{\exercise[(opt)]} command initiates an exercise with the title \texttt{Exercise}, typeset as a document subsection, followed by automatic numbering, unique to the entire document. The optional parameter \texttt{(opt)} is utilized to include additional text on the same title line, such as specifying a subject or a marking scheme. Thus, using \texttt{\exercise[(to begin)]} results in:

\begin{center}
\begin{tabular}{|p{\textwidth}|}
\hline
\textbf{Exercise 1 (to begin)}
\hline
Try this first command; easy!
\end{tabular}
\end{center}
To bring optional text closer to the exercise number, you can employ `\unskip` which removes any preceding space. Take a look at the following example, achieved with `\exercise[\unskip*** (difficult)]`:

<table>
<thead>
<tr>
<th>Exercise 2*** (difficult)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Calculate $1 + 1$.</td>
</tr>
</tbody>
</table>

`\exercise` The term “Exercise” is automatically translated into various languages\(^1\) depending on the language that is loaded (via `babel` or `polyglossia`). You can alter it by modifying `\exercisename`. A better approach is to use macros from the `translations` package by Clemens Niederberger [6] (which allows dynamic language switching), e.g. `\DeclareTranslation{swedish}{exesheet-exercise}{"Ovning"}`.

`\labelexercise` This command combines `\exercisename` with the exercise number and can be redefined. For instance, if you want to include a period after the exercise number, you can redefine it as follows:

\begin{verbatim}
\renewcommand{\labelexercise}{\exercisename~\theexercise.}
\end{verbatim}

If you wish to alter only the numbering style, you can redefine the `\theexercise` command based on the `exercise` counter.

`\labelexercisestyle` This macro, which is initially empty, enables the definition of a specific style for exercise titles. In this document, we have set the following in the preamble:

\begin{verbatim}
\renewcommand{\labelexercisestyle}{\rmfamily\color{black}}
\end{verbatim}

`\exercise*` The starred version `\exercise*[(opt)]{\langle label \rangle}` permits the selection of an alternative `\langle label \rangle` for a specific exercise while omitting the numbering. For instance: `\exercise*[(Fermat’s theorem)]{Problem}` results in:

<table>
<thead>
<tr>
<th>Problem (Fermat’s theorem)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Prove that there are no positive integers $x, y, z$ such that $x^n + y^n = z^n$ for any integer $n$ greater than 2.</td>
</tr>
</tbody>
</table>

### 2.2 The `\subpart` command

`\subpart` An exercise may consist of multiple parts, which can be created using the `\subpart[(opt)]` command. The part title is typeset similar to a sub-subsection.

<table>
<thead>
<tr>
<th>Exercise 3</th>
</tr>
</thead>
<tbody>
<tr>
<td>Part A (preliminary)</td>
</tr>
<tr>
<td>To begin, prepare your cup of tea.</td>
</tr>
<tr>
<td>Part B</td>
</tr>
<tr>
<td>Now you are ready to proceed with the current exercise.</td>
</tr>
</tbody>
</table>

\(^1\)Currently, translation is integrated into the package for the following languages: French, German, Spanish, Italian, and Portuguese.

\(^2\)In this document, real section and subsection titles have been highlighted by modifying their color and font (sans serif) using the `\allsectionsfont` macro from the `sectsty` package [9].
The following macros allow customization in the same manner as for \exercise.

\thesubpart  
By default, subpart numbering employs letters: A, B, C, and so on. This numbering style can be modified using the \thesubpart command, which relies on the subpart counter. For example, you can redefine it as follows: \renewcommand\thesubpart{\arabic{subpart}}.

\subpartname
\labelsubpart
\labelsubpartstyle
The \subpart command utilizes \subpartname (with automatic translation in several languages according to the chosen language), as well as \labelsubpart and \labelsubpartstyle, all of which can be modified.

\thesubpart*
Similar to \exercise*, the starred version \subpart*{⟨opt⟩}{⟨label⟩} permits an alternative ⟨label⟩ and omits the numbering. For instance, you can use \subpart*{First part}.

2.3 The \annex command

\annex
The \annex{⟨opt⟩} command composes the title ANNEX in uppercase letters, centered, using the subsection style, with an optional parameter that will be added on the same line.

ANNEX (to be returned)

\annexname
The term “Annex” is automatically translated into several languages (depending on the chosen language). It can be extended to additional languages or altered by redefining \annexname or by utilizing macros from the translations package [6].

\annexstyle
The style of the annex title is determined by the \annexstyle macro, which is defined as follows: \newcommand\annexstyle{\MakeUppercase}. This command may be redefined according to your preferences.

2.4 Titles in the table of contents

[exetoc={bool}]
By default, the titles Exercise, Part and Annex are included in the table of contents, if there is any, or in the PDF file’s summary when the hyperref package is utilized. To prevent this, you can set the package option exetoc=false (with the default being true). However, note that optional title arguments will always be ignored in the table of contents.

2.5 Short exercises: the \exe command

\exe
The \exe command initiates an exercise with the abbreviation Ex. followed by the exercise number. This is achieved without utilizing sectioning commands, and the exercise content begins on the same line. An exercise begins a new paragraph without any indentation.

Ex. 4 — This is a brief exercise that can encompass several paragraphs or questions.

Here for example a new paragraph begins.

Ex. 5 — This is another concise exercise.
The abbreviation **Ex** can be modified by redefining \texttt{\textbackslash exname} or with macros from the `translations` package \[6\]. The `\texttt{\textbackslash exlabel}` macro combines `\texttt{\textbackslash exname}` with a period then the exercise number (given by the same `exercise` counter), while `\texttt{\textbackslash exsepmark}` typesets a long dash. These characteristics can be altered by redefining these commands.

The starred version doesn’t display a separator, as demonstrated below:

\begin{exe*}
\texttt{Ex. 6} Another short exercise without a separator.
\end{exe*}

### 3 Enumerations and lists

#### 3.1 List settings

Enumeration lists are used to represent questions and sub-questions within exercises. To provide clear emphasis, labels are typeset in bold. Additionally, these labels are aligned to the left, positioned at the start of the line without indentation, and the vertical spacing between items is increased compared to standard \LaTeX{} lists. These formatting adjustments are achieved using the `\texttt{\setlist}` command, a feature from the `enumitem` package by Javier Bezos \[2\].

#### Exercise 7

1. First question
   - (a) First sub-question
   - (b) Second sub-question

2. Second question

The `\texttt{\enumerate}` environment takes an optional parameter, that allows, among others things, the typesetting of alternative list labels. For instance, typing \begin{enumerate}[label=\texttt{\textbackslash alph*},font=\texttt{\itshape\normalfont}] will yield the labels “a), b), c)...”. There are many other options available (see the `enumitem` \[2\] package documentation)\[3\]. Label font formatting can be changed globally using `\texttt{\setlist}[\enumerate]{font=...}` (called after `\begin{document}`).

Lists created with the `\itemize` environment retain their default configuration\[4\].

The package option `\texttt{\textbackslash setlist=false}` prevents changes to enumeration lists and reverts to the default \LaTeX{} settings (the default value is `true`).

#### 3.2 List of exercises : the `exenumerate` environment

When an exercise sheet consists of short, independent questions, it might be unreasonable to display the full title `Exercise` for each one. In addition to the previously

\[3\]Labels can also be modified using a “shortlabel” argument, e.g. `\begin{enumerate}[A.], or globally through the redefinition of `\texttt{\textbackslash labelenumi}` or `\texttt{\textbackslash labelenumii}` commands.

\[4\]However, the `french` option of the `babel` package changes the appearance of `itemize` lists and employs long dashes as labels for each list level. This can cause issues when mathematical content follows the dash symbol, as it might be mistaken for the minus sign. Thus, with the option `\texttt{\textbackslash setlist=true}`, the default \LaTeX{} `itemize` list style is reinstated with `\texttt{\frenchsetup{StandardLists=true}}`. 

---

5
mentioned `\exe` command, we offer an even more streamlined solution using the `exenumerate` environment. This environment is essentially an enumeration list with increased spacing between items, compared to the `enumerate` environment. Here is an example (the main list uses the `exenumerate` environment, while the sub-list is created using the standard `enumerate` environment):

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 sentence in French: Cum sociis natoque penatibus et magnis dis parturient montes, nascetur ridiculus mus.

The `exenumerate` environment (also based on the `enumitem` package) accepts an optional parameter, similar to the `enumerate` environment.

### 3.3 Items aligned by row: `tablenum1`, `tablenuuma`, `tablitem`

These three environments are employed to typeset brief questions (`tablenum1`), sub-questions (`tablenuuma`) or `itemize` lists (`tablitem`) on the same line. They share the same syntax: `\begin{tablenum1}{(opt)}{(cols)}`. The `(cols)` parameter denotes the number of columns utilized by the environment. It must be enclosed in parentheses. This parameter can be omitted, in which case its default value is 2. Similar to conventional lists, each item is initiated with the `\item` command.

Internally we have utilized the `\NewTasksEnvironment` macro from the `tasks` package by Clemens Niederberger [3]. The usage of the optional argument `(opt)` is explained in the documentation of this package. For example, similar to the `enumitem` package [2], `label=`arabic*) produces an Arabic numbering followed by a closing parenthesis. Additionally there are numerous possibilities for arranging items in original ways. For instance, the `\item*` command allows you to specify the number of columns the item is supposed to span. In the subsequent example, the five `\item` commands are sequentially positioned between `\begin{tablenum1}(3)` and `\end{tablenum1}`. Notice that numbering occurs line by line in this context.

<table>
<thead>
<tr>
<th>Exercise 8</th>
</tr>
</thead>
<tbody>
<tr>
<td>Calculate the derivative of the following functions:</td>
</tr>
</tbody>
</table>

1. \( f(x) = \frac{1 - x^2}{e^x + e^{-x}} \),
2. \( g(x) = \ln \left( \frac{1 - x}{1 + x^2} \right) \),
3. \( h(x) = \int_0^1 e^{xy} \, dy \),
4. \( k(x) = \sum_{i=1}^{\infty} \frac{1}{x^i} \),
5. \( l(x) = \int_{\frac{1}{2}}^{x} \frac{1}{\ln t} \, dt \).
For \texttt{tablenum}, labels are letters, \(a, b, c, \ldots\), enclosed in parentheses.

\texttt{labelenumone} \\
\texttt{labelenuma}

You can change the labels by redefining the macros \texttt{\renewcommand\labelenuma()} (for \texttt{tablenum1}) and \texttt{\labelenuma()} (for \texttt{tablenuma}), using the \texttt{task} counter: e.g. \texttt{\renewcommand\labelenuma{\Alph{task}.}} yields the labels \(A, B, \ldots\)

If the \texttt{exesheet} package is invoked with the option \texttt{setlist=false}, labels within \texttt{tablenum1} and \texttt{tablenuma} environments will be presented with indentation and in normal font rather than bold. You can change the label formatting globally with the command \texttt{\settask}, e.g. \texttt{\settask{label-format=\itshape}}. You can also completely redefine the environments using \texttt{\RenewTasksEnvironment}. When \texttt{setlist=true}, place these commands after \texttt{\begin{document}}.

When you intend to utilize \texttt{tablenuma} (or \texttt{tablitem}) immediately after inserting the \texttt{\item} command within an \texttt{enumerate} environment, a vertical misplacement may occur. To achieve proper vertical spacing in such cases, we offer the starred environments \texttt{tablenuma*} and \texttt{tablitem*}, with corrected alignment as shown below:

\begin{enumerate}
  \item \texttt{\texttt{(a)}} One \hfill \texttt{(b)} Two \hfill \texttt{(c)} Three.
\end{enumerate}

If the vertical alignment is still not perfect, include \texttt{\mbox{\vspace{⟨height⟩}}} just after \texttt{\item} and before invoking \texttt{\begin{tablenuma*}} (or \texttt{\begin{tablitem*}}), where \(⟨height⟩\) can be a positive or negative length.

### 3.4 Items aligned by column: \texttt{colsenum, colsitem}

To achieve numbering of items by column, we provide the \texttt{colsenum} environment: \texttt{\begin{colsenum}{⟨opt⟩}{⟨cols⟩}}. The mandatory parameter is the number of columns, and the optional parameter will be passed to the underlying \texttt{enumerate} environment, allowing you to change the numbering type (e.g. \(a, A, \ldots\)), among other possibilities. To use this environment, you need to load the \texttt{multicol} package in the preamble. Here’s an example with \texttt{\begin{colsenum}{3}}:

**Exercise 9**

Calculate the derivative of the following functions:

1. \(f(x) = \frac{1 - x^2}{e^x + e^{-x}}\)
2. \(g(x) = \ln(\frac{1 - x}{1 + x^2})\)
3. \(h(x) = \int_0^1 e^{xy} \, dy\)
4. \(k(x) = \sum_{i=1}^{\infty} \frac{1}{x^i}\)
5. \(l(x) = \int_\frac{1}{2} x \, dt\).

\texttt{colsenum*}

It may be observed that, on each line, items are not necessarily properly aligned, which can result in ungraceful effects. On the other hand, the \texttt{colsenum} environment doesn’t attempt to align columns from the bottom by adjusting the vertical spacing between items. If you desire this alignment (which is the default behavior in \texttt{multicol}), you can use the \texttt{colsenum*} environment (with the same syntax as \texttt{colsenum}). Here’s what we obtain with \texttt{colsenum*}:
Exercise 10

Calculate the derivative of the following functions:

1. \( f(x) = \frac{1 - x^2}{e^x + e^{-x}} \),
2. \( g(x) = \ln \left( \frac{1 - x}{1 + x^2} \right) \),
3. \( h(x) = \int_0^x e^{xy} \, dy \),
4. \( k(x) = \sum_{i=1}^{\infty} \frac{1}{x^i} \),
5. \( l(x) = \int_{\frac{1}{2}}^x \frac{1}{\ln t} \, dt \).

We can observe that these alignments are not as elegant as those achieved through row numbering. However, column numbering might still be more suitable when dealing with numerous items of varying heights, and especially when the number of items can differ from column to column. Additionally, a benefit of \texttt{colenum} is that the label selection is automatic, based on the list level (and the language), unlike \texttt{tablenum1} or \texttt{tablenuma}.

\texttt{colsitem} For \texttt{itemize} lists, the \texttt{colsitem} environment generates items aligned by column, unlike the line-by-line alignment of \texttt{tablitem}. It follows the same syntax as \texttt{colenum}: \texttt{\begin{colsitem}[(opt)]{cols}\end{colsitem}}. The optional parameter, passed to the underlying \texttt{itemize} environment, permits the modification of the item label (bullet by default). Furthermore, just like \texttt{colenum*}, the \texttt{colsitem*} environment produces column alignment from the bottom.

4 Questions and solutions

4.1 Environments questions and answers

The \texttt{exesheet} package offers two environments, \texttt{questions} and \texttt{answers}, which allow you to optionally show or hide questions and answers within exercises.

\[\texttt{[output=(opt)]}\] The output is governed by the \texttt{output} key option which recognizes three values: \texttt{questions}, \texttt{answers}, and \texttt{both}. The \texttt{questions} value shows only questions without answers, \texttt{answers} displays answers without questions, and \texttt{both} (the default option) displays both questions and answers.

\texttt{\correctionstyle} \texttt{correctioncolor} In the default case where both questions and answers are displayed, the answers are typeset using the \texttt{\correctionstyle} style, which utilizes the color \texttt{correctioncolor}. You can modify this color using the \texttt{\definecolor} macro\footnote{The \texttt{\definecolor} command is provided by the \texttt{xcolor} package developed by Uwe Kern, which is automatically loaded by \texttt{exesheet}.}. By default, \texttt{\definecolor{correctioncolor}{rgb}{0,0.2,0.6}} is used, resulting in a kind of dark blue.

\texttt{\correctionname} Furthermore, when using \texttt{output=both} the title \texttt{Correction} is displayed at the beginning of \texttt{answers} environments. This title is defined by the \texttt{\correctionname} macro, with translation available in several languages, and it can also be modified. For instance you might prefer “Solution” over “Correction”. The style defined by \texttt{\correctionstyle} will be applied to the title as well as the entire environment. Here’s an example to illustrate this:
Exercise 11

1. Is the exesheet package useful?

2. Aren’t there any other packages that deal with exercises?

Correction

1. The exesheet package is useful for teachers.

2. There are numerous other packages that handle exercises and provide the capability to create questions and solutions separately. For instance the exercise package by Paul Pichaureau, exercises by Roger Jud, exsheets (now 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, and more. They are briefly presented in section 6.3.

When only answers are displayed, the text color remains black and the word “Correction” is not displayed.

4.2 More about answers environments

Internally, we have utilized the \comment and \endcomment macros from the versions package by Uwe Lück [4]. Moreover, the versions package [4] offers the \excludeversion{⟨env⟩} and \includeversion{⟨env⟩} macros which allow for the exclusion or inclusion of any environment ⟨env⟩. These “optional” environments can be nested.

However the questions and answers environments serve a broader purpose beyond merely displaying or hiding text. You can choose to have a single answers environment for the entire sheet, or alternatively, have separate answers environments for each exercise, exercise part, question, or sub-question. The format in which the title Correction should appear in the output, and its placement in the table of contents or PDF file summary, depends on the nesting level of the environment. In fact, the rendering of the Correction title and its corresponding table of contents level will be automatically calculated by the environment.

However, users might wish to adjust the title’s level themselves. To achieve this, you can manually set the level of the title “Correction” using an optional ⟨level⟩ argument which is defined as follows: 1 for section-level titles, 2 for subsections (akin to Exercise), 3 for sub-subsections (similar to Part), other numbers for lower levels (which won’t appear in the table of contents or in the PDF file’s summary).

Caution should be taken that, if the questions environment is not used beforehand in the same exercise (or part), the answers environment will consider the correction as global for the entire sheet (or exercise) and will reset the exercise (or part) counter. This can be managed properly with the optional argument. For example, use \begin{answers}[2] to prevent exercise counter reset, or \begin{answers}[3] to prevent subpart counter reset.

The starred version answers* doesn’t display the Correction title.
4.3 Commands \question, \answer and \answerspace

\question Instead of using questions and answers environments, we can also employ the simpler \question{(ques)} and \answer{(ans)} macros. The visibility of \{ques\} and \{ans\} content is regulated by the same previous output=\{opt\} key option. This approach might be more fitting when you wish to display answers immediately after each question item. The title “Correction” won’t appear at the start of each answer with the \answer macro. The answers are also formatted using \correctionstyle if output=both. However these commands do not support verbatim text within them, unlike the questions and answers environments.

\question* When a code must be executed only when questions are displayed but not answers, or the contrary, you have the starred versions e.g. \question*{\pagebreak}.

\answer* Some teachers are accustomed to providing their students with documents where questions are typeset, leaving blank spaces instead of answers. This layout allows students to fill in their responses on the paper. To achieve this, thanks to a suggestion from Maxime Chupin, we offer the \answerspace{(\text{height})} macro, in which the parameter \{height\} is a valid length, e.g. \answerspace{3cm}.

[\answerspace=(\text{bool})] The blank spaces introduced by \answerspace can be displayed or hidden, controlled by the \answerspace option key, which can be set to either true or false (the default). The \answerspace key option has no effect (equivalent to false) when the answers are displayed (output=answers or both). Of course the \answerspace macro is not meant to be used within answers environments.

5 Marking scheme commands

The exesheet package provides several commands to display a marking scheme, with optional comments and explanations about answers in the margins.

5.1 The \points command

\points The \points{(pts)} command displays the number of points awarded for an exercise. It is intended to be included in the optional argument of the \exercise command\(^6\). In the following example, we used \exercise{\points{5}}:

<table>
<thead>
<tr>
<th>Exercise 12</th>
<th>5 points</th>
</tr>
</thead>
<tbody>
<tr>
<td>Try to read this document to the end without drinking tea and you get five points.</td>
<td></td>
</tr>
</tbody>
</table>

When only the answers are displayed in an exercise, the \points macro doesn’t show the points. Further, we provide another macro, which displays points in questions like here, and differently in answers environments (see section 5.5).

\pointsname The term “points” (or “point” in the singular if \{pts\} is less than 2) is appended and is automatically translated into several languages (and can also be modified).

\pointsname You can adjust the \points command’s style through \pointsname. The color setting (red by default) is managed by \pointsname using \definecolor, for example you can declare: \definecolor{pointscolor}{named}{blue}.

\pointsstyle \pointsstyle \pointscolor

---

\(^6\)However using \points in the optional argument of \exercise is not compatible with the memoir class, as the memoir class redefines section commands.
5.2 The \pts command

The \exe macro or as a list with the \exenum environment, the marking scheme can be shown in the margin, aligned with the line where the \pts\{num\} command is placed (typically the first line of the exercise). The \langle num \rangle parameter represents the number of points assigned to the exercise. Here’s an example with \exe\pts\{3\}... \exe\pts\{1.5\}...

<table>
<thead>
<tr>
<th>(3 pts)</th>
<th>Ex. 13 — The first short exercise with a marking scheme.</th>
</tr>
</thead>
<tbody>
<tr>
<td>(1.5 pt)</td>
<td>Ex. 14 — The second one.</td>
</tr>
</tbody>
</table>

\ptsname The abbreviation “pts” (or “pt” when the number of points is less than 2) is added automatically using \ptsname macros (translated in several languages if babel or polyglossia is loaded). The point’s display color is defined by ptscolor, changeable via \definecolor (red by default). The display style is determined by \ptsstyle, which among other things, adds parenthesis around.

\[\text{display=(opt)}\] The marking scheme visibility is controlled by the display option key. The default option is display=none, keeping the marking scheme hidden. To reveal the marking scheme, use display=pts. More details are available in section 5.4.

\[\text{marginpos=(opt)}\] The positioning of the scale is determined by the marginpos option key, typically left or right. The default value is left even though LATEX positions marginal notes on the right side by default. This option has no impact when display=none.

For a two-sided document, the default behavior is to place text in the outer margin, which is wider than the inner margin (that contains the binding). The outer margin is positioned on the right side on odd pages and on the left side on even pages. Therefore, the marginpos option can also take the values inner or outer. If you specify left or right when the twoside mode is activated, this value will be converted to outer, accompanied by a warning message.

With the twoside mode, marginal notes might occasionally appear on the wrong side of a page. This is a known LATEX bug, and the solution involves using the mparhack package by Tom Sgouros and Stefan Ulrich [8] (which exesheet automatically includes for documents in two-side mode) and running E\LaTeX\ twice. If necessary, a warning message will prompt you to perform the re-run.

5.3 Commands \totalexe, \note* and \note

For a more comprehensive marking scheme, the following commands are available.

\totalexe The \totalex\{num\} macro displays the total number of points of an exercise. By default, it appears inside an oval box, with the addition of the word “pts” (or “pt”) in bold red. In the following example, the exercise title has been generated using \exercise[\totalexe\{4\}].

\note* For each answer or solution in the correct version, the \note*\{\langle num \rangle\} command indicates the number of points allocated to that question. The appearance slightly varies compared to \pts: by default the number is displayed in bold without the “pts” or “pt” suffix, and without parenthesis. In the following example, for answer 3, we employed \note*\{1.5\}, placed right after \item.
The \note{⟨comment⟩} macro is utilized to provide additional information regarding the marking scheme and to explain how points are assigned. In the \langle comment \rangle argument you can use \textbackslash \ to create a line break or even \textbackslash[⟨height⟩] to adjust the line spacing by ⟨height⟩.

Placing \note*[⟨num⟩]{⟨comment⟩} at the beginning of an answer is often practical. In such cases LaTeX will align the margin notes vertically, which leads to a warning like: LaTeX Warning: Marginpar on page ... moved. However, this warning is not an issue, as LaTeX can usually handle the arrangement of these marginal notes, stacking them one below the other. Nonetheless, to prevent unnecessary warnings, you can combine both commands into a single one by specifying the number of points as an optional argument of the \note command: \note[⟨num⟩]{⟨comment⟩}.

The initial comment in the following example is generated (immediately after \item) using \note[1]{0.5 for the anti-derivative\0.5 for simplifying}.

\section*{Exercise 15}

For each subsequent question, determine whether the statement is true or false. Provide a thorough justification for your answer.

1. \[\int_{0}^{\sqrt{3}} \frac{1}{x + \sqrt{3}} \, dx = \ln 2,\]
2. \[\int_{2}^{e} \frac{1}{x \ln x} \, dx = -\ln 2,\]
3. The function \(F\), defined on \(\mathbb{R}\) by \(F(x) = \int_{0}^{x} \frac{1}{t^2 + t + 1} \, dt\), is increasing on \(\mathbb{R}\).

\section*{Correction}

1. We calculate:
\[
\int_{0}^{\sqrt{3}} \frac{1}{x + \sqrt{3}} \, dx = \left[ \ln \left( x + \sqrt{3} \right) \right]_{0}^{\sqrt{3}} = \ln \left( 2\sqrt{3} \right) - \ln \sqrt{3} = \ln \left( \frac{2\sqrt{3}}{\sqrt{3}} \right) = \ln 2.
\]

\text{TRUE.}

2. We have \(\frac{1}{x \ln x} = \frac{1}{\ln x} = \frac{u'(x)}{u(x)}\) with \(u(x) = \ln x\), which is positive on \([2, e]\). Hence
\[
\int_{2}^{e} \frac{1}{x \ln x} \, dx = \left[ \ln(\ln x) \right]_{2}^{e} = \ln(\ln e) - \ln(\ln 2) = \ln 1 - \ln(\ln 2) = -\ln(\ln 2).
\]

\text{FALSE.}

3. The function \(F\), defined on \(\mathbb{R}\) by
\[
F(x) = \int_{0}^{x} \frac{1}{t^2 + t + 1} \, dt,
\]
is derivable on \(\mathbb{R}\) 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 \(\mathbb{R}\).

\text{TRUE.}
In the comment for answer 2, a larger vertical space is created with the optional argument `\[2ex]\` for line break. The last comment, which isn’t positioned next to the points number, was produced by placing the following on the first line after the formula: `\note{0.5 for $F'$\$ackslash\!1$ for the sign of $F'$ and conclusion}`.

The color and style for displaying points in `\totalex` and `\note*` can be customized using `\markingcolor` and `\markingstyle`, respectively. The oval box produced by `\totalex` is created using the `\ovalbox` command from the `fancybox` package by Timothy Van Zandt [5], with corner arcs set by `\cornersize{1}`. The box’s length is determined by `\ptsboxlength`, and not by the box’s content, to ensure uniformity across exercises.

By default, comment notes are typeset in a dark green color defined by `\definecolor{notecolor}{rgb}{0.0,0.4,0.0}`. The style of comments is determined by the `\notestyle` macro.

### 5.4 Margin notes options

The `\display{opt}` key option governs the presentation of the marking scheme: as discussed previously (subsection 5.2), `\display=none` shows nothing. When using `\display=pts` the numbers provided as arguments to `\pts`, `\totalex`, `\note*` or as optional arguments of `\note{⟨num⟩}{…}` will be exhibited. The final option is `\display=notes` which reveals the complete marginal notes, containing points and comments (the mandatory argument of `\note`), as illustrated in the previous example.

As previously mentioned in subsection 5.2, the side on which to position the scale is determined by the `\marginpos{opt}` key option, with possible values of `left` and `right` (or `inner` and `outer` if the document is in `twoside` mode).

The margin layout is governed by the `\marginwidth{opt}` key option, which can take one of the following values: `standard`, `expand`, or `unset`. This option has no effect when `\display=none`. In this case, both the left and right margins have the same width, except in a two-sided document where the ratio between the left and right margins is 2:3. Otherwise the `\marginwidth` key option behaves as follows:

- **standard** The left margin is widened, and the right margin is reduced, with a ratio of 3:2 (or 2:3 if `marginpos=right`). The text body is shifted without changing its width. The margin paragraph width remains relatively short (depends on page geometry). This option is not ideal for lengthy comments.

- **expand** (default value) The behavior is the same as with the `standard` value when `\display=pts`. However, when `\display=notes`, the margin expands with a ratio of 3:1 (or 1:3) and the width of margin paragraphs increases.

- **unset** This option is provided for cases where the previous settings are not suitable. In this case, no adjustments are made to the margin width. Instead, you can define your own settings using the convenient `\geometry` macro from the `geometry` package by Hideo Umeki [1]. For instance, you can place the following in the preamble:

  `\geometry{hmarginratio=2:1,marginparwidth=2.5cm}`.

  If `marginpos=right`, you need to invert the ratio, e.g. 1:2 instead of 2:1. If `marginwidth` is not set to `unset`, such a command will have no effect.
Margin settings are applicable to the entire document and need to be configured in the preamble.

The package option `noteragged` controls the text alignment within the margins for the mandatory argument of \note. It offers the following values: left, right, center, justify or twoside. The default value is `noteragged=left`, resulting in right-aligned text, which is common for text in the left margin. When `noteragged=right`, the text is left-aligned. Using `justify` makes the text justified, aligning with \TeX{}'s default behavior for marginal notes. Finally `noteragged=twoside` aligns text to the left on odd pages and to the right on even pages in a two-sided document. It has no effect otherwise (the default `noteragged=left` is used and a warning message appears in the terminal).

When `display` is not set to notes, the `noteragged` option has no impact, as it specifically applies to text within the mandatory argument of \note.

5.5 The \totalpoints command

The `\totalpoints{⟨num⟩}` macro serves as a replacement for `\points` when using a comprehensive marking scheme. When the scale is not displayed, it functions similarly to `\points` (visible in questions but not in answers), and when the scale is shown, it's akin to `\totalexe`. For instance, in exercise 15, we could have used `\totalpoints` instead of `\totalexe`. Thus, if the detailed marking scheme is not displayed, the total points would be presented similarly to exercise 5.1.

5.6 Marking scheme consistency checking

The marking scheme can be checked out\footnote{Thanks to Denis Bitouzé for his suggestion about this feature.} using the key-val option `checkpts=true` (or just `checkpts`); the default value is `false`.

For each exercise, the cumulative points allocated to each question (via `\pts`, `\note*` or `\note[ ]`) are compared to the exercise’s total specified in `\points`, `\totalexe` or `\totalpoints`. A warning message will be displayed in the shell to indicate whether the scale is valid for the exercise or not. For example:

```
Package exesheet warning: Exercise 3: Sum of points is 4.5pt instead of 5pt.
```

Both comma notation (e.g. 4,5) and decimal point format (e.g. 4.5) may be accepted, depending on your chosen language. The control is made at the beginning of the subsequent exercise, inside the `\points`, `\totalexe` or `\totalpoints` macros. No deep checking will be processed at this level if no points are displayed for the questions inside the exercise (with `display=none` option).

```
\totalsheet
```

At the end of the document, the last exercise is checked, followed by a global examination of the entire sheet. This last task requires knowledge of the total points for the sheet, which must be given by the `\totalsheet{⟨points⟩}` macro in the preamble; otherwise, a warning message will be displayed. If subtotals have been assigned to exercises and `displayed`, the overall comparison is made between the sum of these subtotals and the total points recorded using `\totalsheet`. If not, the evaluation encompasses the sum of points for each individual question. A subsequent warning message indicates the outcome of this last verification. Finally, a message indicates whether all scale controls have been successfully passed or not.
6 Options and comparison with other packages

6.1 Summary of available options

Here we provide a summary table of the available options. Details on their usage can be found in the respective sections. The default value is displayed in bold.

<table>
<thead>
<tr>
<th>Key</th>
<th>Possible values</th>
<th>See section</th>
</tr>
</thead>
<tbody>
<tr>
<td>exetoc</td>
<td>true, false</td>
<td>2.4</td>
</tr>
<tr>
<td>setlist</td>
<td>true, false</td>
<td>3.1</td>
</tr>
<tr>
<td>output</td>
<td>questions, answers, both</td>
<td>4.1</td>
</tr>
<tr>
<td>answerspace</td>
<td>true, false</td>
<td>4.3</td>
</tr>
<tr>
<td>display</td>
<td>none, pts, notes</td>
<td>5.2, 5.4</td>
</tr>
<tr>
<td>marginpos</td>
<td>left (inner), right (outer)</td>
<td>5.2, 5.4</td>
</tr>
<tr>
<td>marginwidth</td>
<td>standard, expand, unset</td>
<td>5.4</td>
</tr>
<tr>
<td>noteragged</td>
<td>left, right, center, justify, twoside</td>
<td>5.4</td>
</tr>
<tr>
<td>checkpts</td>
<td>true, false</td>
<td>5.6</td>
</tr>
<tr>
<td>correct</td>
<td>true, false, conditional</td>
<td>see below</td>
</tr>
</tbody>
</table>

When an invalid key is provided, an error is generated. However, an unrecognized value only triggers a warning message:

Value ... is not supported by ... option on input line ...

For each option, you can set them through the class or package invocation, e.g.

\usepackage[output=answers,display=notes,noteragged=right]{exesheet}

\exesheetset{list of ⟨key⟩=⟨value⟩} command. Note that some options, output, answerspace, display, and noteragged, can be changed dynamically, even within the document, while the others are applicable in the preamble exclusively. Dynamic options are processed with each call, whereas the others are processed once, at the beginning of the document.

A special option, correct, can be employed when using the exesheet class or in conjunction with the schooldocs package. This option adds “Correct version” (or its translation) to the document title and headers. Possible values are: true, false (by default) or conditional. Using correct=conditional, it behaves as true when answers are displayed and false when they’re not.

6.2 Alternative commands

Prior to version 2.0, we used specialized commands to configure output and display options. Thanks to a suggestion from Maxime Chupin and Denis Bitouzé, we have now implemented key=value options. Although the latter is more user-friendly, one may prefer the old commands, so they are still supported, but will trigger a warning message. They are presented below.

However, the previous options nosetlist and notoc are no longer supported.

\questionsonly The command \questionsonly is equivalent to setting output=questions
\answersonly and \answersonly means output=answers.
\displaypts The commands \displaypts and \displaypoints are equivalent to setting display=pts.
\texttt{\textbackslash displaynotes} means \texttt{display=notes}, and \texttt{\textbackslash displaynotesright} corresponds to \texttt{display=notes,marginpos=right}. These two commands have an optional argument \texttt{\textbackslash displaynotes\{\textbackslash ragged\}} where \texttt{\textbackslash ragged} is an alignment command to work inside margin notes. By default it is \texttt{\textbackslash RaggedLeft} with \texttt{\textbackslash displaynotes} and \texttt{\textbackslash RaggedRight} with \texttt{\textbackslash displaynotesright}.

6.3 Comparison with other packages

In this section, we will provide an overview of the functionalities (as of today December 30, 2023) of various packages or classes found in the 'Exercise' or 'Exam' sections of the CTAN archives (Comprehensive \TeX{} Archive Network). Considering the substantial number of packages in these sections, some omissions may have been unintentionally made. Those excluded are those with documentation not in English or primarily dedicated to producing multiple-choice questions or random question generation. Are also excluded those with non-free license. We have focused here on typesetting functionalities and not on managing exercise databases as there are specialized packages or external softwares for that.

The following table is not a result of tests but presents a summary of information obtained from the documentation of these packages.

<table>
<thead>
<tr>
<th>Functionality</th>
<th>A</th>
<th>B</th>
<th>C</th>
<th>D</th>
<th>E</th>
<th>F</th>
<th>G</th>
<th>H</th>
<th>I</th>
<th>J</th>
<th>K</th>
<th>L</th>
<th>M</th>
<th>N</th>
<th>O</th>
<th>P</th>
</tr>
</thead>
<tbody>
<tr>
<td>optional text for exercise titles</td>
<td>X</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>subparts of exercises</td>
<td></td>
<td>X</td>
<td>X</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>X</td>
</tr>
<tr>
<td>annex title</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>X</td>
<td>X</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>exercise titles in TOC of PDF files</td>
<td></td>
<td>X</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>X</td>
<td>X</td>
<td>X</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>short labels for exercises</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>X</td>
<td>X</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>hiding questions or answers*</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>different placements for answers</td>
<td>X</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>X</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>changing answers placement in output</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>X</td>
<td>X</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>blank spacing in place of answers</td>
<td></td>
<td>X</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>X</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>marking scheme commands</td>
<td></td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>X</td>
<td>X</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>various positions of points</td>
<td></td>
<td>X</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>marking scheme calculation/checking</td>
<td></td>
<td>X</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>X</td>
<td></td>
<td></td>
</tr>
<tr>
<td>detailed notes for scoring guide</td>
<td>X</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>X</td>
<td></td>
</tr>
</tbody>
</table>

*These commands come from the \texttt{ragged2e} package by Martin Schröder [7].
7 Implementation

7.1 Options and required packages

The exesheet class is built upon the article class and transfers all its unknown options to it. The use of \ProcessKeyvalOptions* is unnecessary within the class as it will be managed by the package.

\begin{verbatim}
\@ifclassloaded{exesheet}{}{\RequirePackage{kvoptions}
\DeclareBoolOption[true]{exetoc}
\DeclareBoolOption[true]{setlist}
\DeclareStringOption[both]{output}
\DeclareStringOption[none]{display}
\DeclareBoolOption[false]{ answerspace}
\DeclareStringOption[left]{marginpos}
\DeclareStringOption[expand]{marginwidth}
\DeclareStringOption[left]{noteragged}
\DeclareBoolOption[false]{checkpts}
\DeclareStringOption[false]{correct}
}\ProcessOptions \relax
\LoadClass{article}
\RequirePackage{exesheet}
\RequirePackage{schooldocs}
\end{verbatim}

Options are defined using the kvoptions package. String options are managed through distinct processing macros that are implemented in their respective sections. For options whose effects cannot be dynamically altered and must be configured in the preamble, they are processed once, at \begin{document}. The other options are executed when this package is loaded (at the end of the package, as \exs@process... commands are not recognized at the outset).

A distinct case is to mention with setlist when utilized in conjunction with babel-french. In this instance, this option is processed immediately (further clarification follows below).

\begin{verbatim}
\@ifclassloaded{exesheet}{}{\RequirePackage{kvoptions}
\DeclareBoolOption[true]{exetoc}
\DeclareBoolOption[true]{setlist}
\DeclareStringOption[both]{output}
\DeclareStringOption[none]{display}
\DeclareBoolOption[false]{ answerspace}
\DeclareStringOption[left]{marginpos}
\DeclareStringOption[expand]{marginwidth}
\DeclareStringOption[left]{noteragged}
\DeclareBoolOption[false]{checkpts}
\DeclareStringOption[false]{correct}
}\ProcessOptions \relax
\LoadClass{article}
\RequirePackage{exesheet}
\RequirePackage{schooldocs}
\end{verbatim}

\PackageInfo{exesheet}{The options ‘notoc’ and ‘nosetlist’}
37 \MessageBreak are no longer supported\gobble}
38 \% \gobble suppresses the line number here
39
40 \def\exs@process@dynoptions{
41 \exs@process@output
42 \exs@process@display
43 \exs@process@noteragged
44 } \% answerspace do not need a special process macro
45
46 \AtEndOfPackage{\exs@process@dynoptions}
47 \AtBeginDocument{
48 \newif\ifexesheet@multicol
49 \@ifpackageloaded{multicol}{
50 \exesheet@multicoltrue\{\exesheet@multicolfalse\}
51 \% configuring the rule color within answers environments
52 \exs@process@setlist
53 \exs@process@marginpos
54 \exs@process@marginwidth
55 \exs@process@checkpts
56 \exs@process@correct
57 \DisableKeyvalOption[\text{action=warning},\text{package=exesheet}]{exesheet}{setlist}
58 \DisableKeyvalOption[\text{action=warning},\text{package=exesheet}]{exesheet}{marginpos}
59 \DisableKeyvalOption[\text{action=warning},\text{package=exesheet}]{exesheet}{marginwidth}
60 \DisableKeyvalOption[\text{action=warning},\text{package=exesheet}]{exesheet}{checkpts}
61 \DisableKeyvalOption[\text{action=warning},\text{package=exesheet}]{exesheet}{correct}
62 }
63
64 \exesheetset The \exesheetset macro can accept key-val options and can be utilized anywhere in the document to adjust certain settings. However, it won’t affect non dynamic options if called outside the preamble. In such cases a warning message occur due to the use of \DisableKeyValOption.
65
66 \def\exesheetset#1{\setkeys{exesheet}{#1}\exs@process@dynoptions}
67
68 The following old macros (used before version 2.0) provide an alternative to keyval options. They are kept for compatibility reasons.
69
70 \newcommand{\questionsonly}{
71 \PackageWarning{exesheet}{Old command \string\questionsonly\space
72 is used. \MessageBreak
73 It can be replaced by the option ‘output=questions’}
74 \renewcommand{\exesheet@output}{\text{output}(\text{questions})}
75 \exs@process@output
76 }
77
78 \newcommand{\answersonly}{
79 \PackageWarning{exesheet}{Old command \string\answersonly\space
80 is used. \MessageBreak
81 It can be replaced by the option ‘output=answers’}
82 \renewcommand{\exesheet@output}{\text{output}(\text{answers})}
83 \exs@process@output
84 }
85
86 \newcommand{\displaypts}{\%
87 \PackageWarning{exesheet}{Old command \string\displaypts\space
88 is used. \MessageBreak
89}
It can be replaced by the option ‘display=pts’
\renewcommand\exesheet@display{pts}
\exs@process@display
\newcommand\displaypoints{%
\PackageWarning{exesheet}{Old command \string\displaypoints\space
is used. \MessageBreak
It can be replaced by the option ‘display=pts’}
\renewcommand\exesheet@display{pts}
\exs@process@display
\newcommand\displaynotes[1][1]\RaggedLeft{%
\PackageWarning{exesheet}{Old command \string\displaynotes\space
is used. \MessageBreak
It can be replaced by the option ‘display=notes’}
\renewcommand\exesheet@display{notes}
\exs@process@display
\renewcommand\noteragged{#1}
\newcommand\displaynotesright[1][1]\RaggedRight{%
\PackageWarning{exesheet}{Old command \string\displaynotesright\space
is used. \MessageBreak
It can be replaced by the option ‘display=notes, margin=right’}
\renewcommand\exesheet@display{notes}
\exs@process@display
\renewcommand\exesheet@margin{right}
\renewcommand\noteragged{#1}
\newcommand\noteragged{#1}

Now, we load several packages. If the geometry package is already loaded, it
will not be reloaded to prevent an option clash. The shortlabel option in the
enumitem package [2] allows the use of labels similar to the enumerate package
such as 1., a), A., and so on. The mparhack package by Tom Sgouros and Stefan
Ulrich [8] is loaded exclusively for documents in twoside mode.
\RequirePackage{ifthen}
\@ifpackageloaded{geometry}{}{\RequirePackage{geometry}}
\RequirePackage{xcolor}
\RequirePackage{shortlabels}{enumitem}
\RequirePackage{tasks}[2020/08/19]
\RequirePackage{versions}
\RequirePackage[fancybox]
\RequirePackage{translations}
\RequirePackage{ragged2e}
\ifthenelse{\boolean{@twoside}}{\RequirePackage{mparhack}}{}

7.2 Internationalization
Here we define keywords along with their translations in French, German, Spanish
Italian, Portuguese. We achieve this using macros from the translations package
by Clemens Niederberger [6]. This package automatically detects the language
being used, as loaded by babel or polyglossia.
7.3 Titles

The exercise counter assigns numbers to exercises throughout the entire document, regardless of sections. To reset the counter manually, simply use \setcounter{exercise}{0}. For an automatic reset at each new section, include the following code in the preamble:
\makeatletter \@addtoreset{exercise}{section} \makeatother.

The parts counter (subpart) depends on the exercise counter and is reset with each new exercise.

The commands \labelexercisestyle and \labelsubpartstyle are initially empty, but they allow you to customize the styling. For example:
\renewcommand\labelexercisestyle{\sffamily}.

The \exe@label macro, which needs the exe@check counter, will be used inside warning messages about the marking scheme (see section 7.6).

By default, the table of contents includes both exercises and parts titles, as controlled by the boolean \ifexesheet@exetoc. To only display exercise titles in the table of contents while omitting parts, include the following code in the preamble: \setcounter{tocdepth}{2}.
7.4 Enumerations and lists

The \setlist command is part of the enumitem package (\setenumerate is deprecated). By default, itemsep=1ex is set for first-level lists, and leftmargin=1.5em is used to align labels with the start of lines.

When using the babel package with the french option, itemize lists are altered to use the same dash label for each list level. These modifications are undone here to revert to the default LaTeX itemize lists, including labels and spaces. We have created the \standardfrenchlists command, which should be invoked within the \AtBeginDocument command or immediately, depending on whether exesheet is loaded before or after babel.

\newcommand\standardfrenchlists{%
\ifpackagewith{babel}{french}{
 \frenchsetup{StandardLists=true}
}{
}}
\ifexesheet@setlist
\standardfrenchlists
% must be executed here (and not at begin doc) if loaded after babel
\fi

\newcommand\labelenumone{\arabic{task}.}
\newcommand\labelenuma{\alph{task})
The `\NewTasks` command is part of the `tasks` package [3]. It enables the definition of the environments `tablenum1`, `tablenuma` and `tablitem`. Horizontal spacing is adjusted to ensure proper alignment with items in other `enumerate` (or `itemize`) environments.

The starred environments `tablenuma*` and `tablitem*` are designed to be employed within an `enumerate` environment, precisely at the outset of an `\item`, in order to achieve correct horizontal alignment. The length of \(-1.667\baselineskip\) has been tested with various font families and sizes. The alignment is generally good.

```latex
% The environment 'tablenum' is deprecated
\PackageInfo{exesheet}{The environment 'tablenum' is deprecated and has been replaced by 'tablenum1'@gobble}
% @gobble suppresses the line number here
```
7.5 Questions and answers

The booleans `exesheet@questions` and `exesheet@answers` govern the visibility of their corresponding environments. These booleans are configured through the `output` key option within the \exs@process@output macro.
\def\exs@process@output{
\ifthenelse{\equal{\exesheet@output}{questions}}{
\setboolean{exesheet@questions}{true}
\setboolean{exesheet@answers}{false}
}{% else if
\ifthenelse{\equal{\exesheet@output}{answers}}{
\setboolean{exesheet@questions}{false}
\setboolean{exesheet@answers}{true}
\exesheet@answerspacefalse
}{% else if
\ifthenelse{\equal{\exesheet@output}{both}}{
\setboolean{exesheet@questions}{true}
\setboolean{exesheet@answers}{true}
\exesheet@answerspacefalse
}{% else
\PackageWarning{exesheet}{Value '\exesheet@output'
is not supported by 'output' option}
}}
}}

We utilize the \texttt{versions} package developed by Uwe Lück [4], which introduces the macros \texttt{\comment} and \texttt{\endcomment}. These macros facilitate conditional displays, a technique also employed in the \texttt{verbatim} and \texttt{version} packages. Additionally, the notable \texttt{codesection} package offers the capability to enclose optional code between \texttt{\BeginCodeSection{⟨\texttt{skip}⟩}} and \texttt{\EndCodeSection{⟨\texttt{skip}⟩}} macros, both in the text body and the preamble. However, these macros cannot be used within an environment as we have done here with \texttt{\comment} and \texttt{\endcomment}. Several of our tests use the \LaTeX{} syntax \texttt{\ifthenelse{\bolean{...}}} since \texttt{\comment} and \texttt{\endcomment} can sometimes interfere with the \LaTeX{} structure \texttt{\if \ldots \else \ldots \fi}.

The two counters \texttt{exe@ini} and \texttt{subpart@ini} are employed in the subsequent \texttt{\set@toclevel} macro.

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

\newcounter{@toclevel}

The internal macro \texttt{\set@toclevel} calculates the title level (counter \texttt{toc@level}) to ensure correct typesetting of “Correction” at the start of an \texttt{answers} environment, when \texttt{questions} and \texttt{answers} are displayed together. It involves comparing the \texttt{exercise} and \texttt{subpart} counters with their values at the time of the \texttt{questions} environment call. The \texttt{@enumdepth} counter indicates the current \texttt{enumerate} list level (with 0 indicating outside of any list). The optional parameter of the \texttt{answers} environment permits the explicit specification of this title level.

\newcounter{@toclevel}
The internal macro \texttt{\typeset@correctionname}, displays the term “Correction” at the appropriate level.

\begin{verbatim}
definecolor{correctioncolor}{rgb}{0,0.2,0.6} % kind of dark blue \newcommand{\correctionstyle}{\color{correctioncolor}} \newcommand{\typeset@correctionname}{ \ifthenelse{\value{@toclevel} = 1}{ \section*{\correctionstyle\correctionname} \ifexesheet@exetoc \addcontentsline{toc}{section}{\correctionname} \fi \setcounter{exercise}{0} }{% else if \ifthenelse{\value{@toclevel} = 2}{% \subsection*{\correctionstyle\correctionname} \ifexesheet@exetoc \addcontentsline{toc}{subsection}{\correctionname} \fi \setcounter{subpart}{0} }{% else if \ifthenelse{\value{@toclevel} = 3}{% \subsubsection*{\correctionstyle\correctionname} \ifexesheet@exetoc \addcontentsline{toc}{subsubsection}{\correctionname} \fi \par \textbf{\correctionstyle\correctionname}\par }\}
\end{verbatim}

Then we proceed to define the \texttt{answers} environment.

\begin{verbatim}
\newenvironment{answers}[1][{}]{% #1 is the optional level \ifthenelse{\boolean{exesheet@answers}}{% \ifthenelse{\boolean{exesheet@questions}}{ \set@toclevel[#1] \typeset@correctionname \correctionstyle \ifexesheet@multicol \renewcommand{\columnseprulecolor}{\color{correctioncolor}} }{% \subsection*{\correctionstyle\correctionname} \ifexesheet@exetoc \addcontentsline{toc}{subsection}{\correctionname} \fi \setcounter{subpart}{0} }{% else if \ifthenelse{\value{@toclevel} = 3}{% \subsubsection*{\correctionstyle\correctionname} \ifexesheet@exetoc \addcontentsline{toc}{subsubsection}{\correctionname} \fi \par \textbf{\correctionstyle\correctionname}\par }%}
\end{verbatim}
When placing \correctionstyle before \subsubsection in the answers environment (as in the case of \typeset@correctionname), the preceding vertical space may become too wide.

\question
\question*
\newcommand{\@question}[1]{\ifexesheet@questions #1\fi}
\newcommand{\@@question}[1]{\ifexesheet@questions\ifexesheet@answers \else #1\fi\fi}
\newcommand{\question}{\@ifstar{\@@question}{\@question}}

\answer
\answer*
\newcommand{\@answer}[1]{\ifexesheet@answers\ifexesheet@questions \else #1\fi\fi}
\newcommand{\@@answer}[1]{\ifexesheet@answers\ifexesheet@questions \else #1\fi\fi}
\newcommand{\answer}{\@ifstar{\@@answer}{\@answer}}
\answerspace
\exs@process@correct

The \answerspace macro was suggested by Maxime Chupin to allow students space for writing their answers on the provided paper.

\exs@process@correct The correct option needs the schooldocs package. It triggers the \correct macro of schooldocs which adds the content of \correctname in the title of the document. Here the option conditional triggers \correct only if output=answers or both.

\def\exs@process@correct{
\ifthenelse{\equal{\exsheets@correct}{false}}{% do nothing
  \% else
  \@ifpackageloaded{schooldocs}{
    \ifthenelse{\equal{\exsheets@correct}{true}}{\correct
      \% else
      \ifthenelse{\equal{\exsheets@correct}{conditional}}{
        \exsheets@answers \correct \fi
      \fi}
    \}{}
  }
}\PackageWarningNoLine{exesheet}{The ‘correct’ option requires \MessageBreak
  \MessageBreak
the ‘schooldocs’ package to be loaded}
}}
}

7.6 Marking scheme options processing

The options display, marginpos, marginwidth and noteragged are handled using the following internal commands.

The display key option determines the value of the two booleans exesheet@pts and exesheet@notes. The exesheet@pts boolean controls the display of the content of \pts and optional arguments of \note, while the exesheet@notes boolean controls mandatory arguments of \note.

\exs@process@display

\exs@process@marginpos

The marginpos key option takes the values left (the default value) or right (or inner and outer). In practice, inner is equivalent to left, but in two-sided mode, the values left or right are converted to outer (which is then the default value for two-sided mode).
The \texttt{marginwidth} option adjusts the ratio between left and right margins based on what needs to be displayed in the margin (points only or full notes)\textsuperscript{9}. When \texttt{display=notes}, the additional length of 1 in corresponds to the default free space to the left of \texttt{oddsidemargin}.

The macros \texttt{\standardmarginwidthfactor} and \texttt{\largemarginwidthfactor} represent the ratios between the total margin width and \texttt{\marginparwidth}.

\begin{verbatim}
\newcommand*{\standardmarginwidthfactor}[1]{
  \addtolength{\marginparwidth}{1in}
  \addtolength{\marginparwidth}{-\marginparsep}
  \setlength{\marginparwidth}{#1\marginparwidth}
}
\newcommand*{\largemarginwidthfactor}[0.8]
\end{verbatim}

\footnote{To ensure the accurate effect on the margin ratio, this option is processed at the beginning of the document, after other commands that could potentially alter the page geometry.}
\addtolength{\marginparwidth}{-\marginparsep}
\addtolength{\marginparwidth}{-1in}
\setlength{\marginparwidth}{1in}

\def\exesheet@smallmargins{
  \geometry{hmarginratio=1:1}
  \leftnotemarginwidth{\standardmarginwidthfactor}
}
\def\exesheet@standardmargins{
  \ifexesheet@leftmargin
    \geometry{hmarginratio=3:2}
    \leftnotemarginwidth{\standardmarginwidthfactor}
  \else
    \geometry{hmarginratio=2:3}
    \rightnotemarginwidth{\standardmarginwidthfactor}
  \fi
}
\def\exesheet@largemargins{
  \ifexesheet@leftmargin
    \geometry{hmarginratio=3:1}
    \leftnotemarginwidth{\largemarginwidthfactor}
  \else
    \geometry{hmarginratio=1:3}
    \rightnotemarginwidth{\largemarginwidthfactor}
  \fi
}
\def\exs@process@marginwidth{
  \ifthenelse{\equal{\exesheet@marginwidth}{standard}}{
    \ifthenelse{\equal{\exesheet@display}{none}}{
      \iftwoside
        \exesheet@standardmargins
      \else
        \exesheet@smallmargins
      \fi
    }{% else display=pts or display=notes
      \exesheet@standardmargins
    }{% else if
      \ifthenelse{\equal{\exesheet@marginwidth}{expand}}{
        \ifthenelse{\equal{\exesheet@display}{none}}{
          \iftwoside
            \exesheet@standardmargins
          \else
            \exesheet@smallmargins
          \fi
        }{% else if
          \ifthenelse{\equal{\exesheet@display}{pts}}{
            \exesheet@standardmargins
          }{% else display=notes
            \exesheet@largemargins
          }
        }{% else if
          \exesheet@standardmargins
        }
    }{% else if
      \exesheet@smallmargins
    }
  }{% else if
    \exesheet@standardmargins
  }
}
For a two-sided document, the \texttt{geometry} package does not correctly set the default width of the margin paragraph; it’s too wide. Therefore, we provide an explicit setting here, which is useful when \texttt{marginwidth=unset}. Otherwise, the setting is handled by the \texttt{marginwidth} key option.

\begin{verbatim}
\if@twoside \rightnotemarginwidth{0.5} \fi
\end{verbatim}

\texttt{\textbackslash exs@process@noteragged} \hspace{1em} \textbackslash exs@process@noteragged} The noteragged option can take one of the following values: \texttt{left}, \texttt{right}, \texttt{center}, \texttt{justify} or \texttt{twoside}. When working with a two-sided document, \texttt{\textbackslash marginpar} can be used with an optional parameter to distinguish left from right contents. In this context, we employ \texttt{\noteraggedleft} and \texttt{\noteraggedright} instead of \texttt{\noteragged}. The \texttt{ragged2e} package by Martin Schröder \cite{ragged2e} offers the commands \texttt{\RaggedLeft}, \texttt{\RaggedRight}, \texttt{\Centering}, and \texttt{\justifying}. These commands yield better results compared to the standard \texttt{\raggedleft}, \texttt{\raggedright} and \texttt{\centering} commands. Margin paragraphs are justified by default in \LaTeX. 

\begin{verbatim}
\newcommand{\noteragged}{\{}\newcommand{\noteraggedleft}{\{}\newcommand{\noteraggedright}{\{}\def\exs@process@noteragged{\ifthenelse{\equal{\exesheet@noteragged}{left}}{\if@twoside \renewcommand{\noteraggedleft}{\RaggedLeft} \renewcommand{\noteraggedright}{\RaggedLeft} \else \renewcommand{\noteragged}{\RaggedLeft} \fi}{% else if \ifthenelse{\equal{\exesheet@noteragged}{right}}{\if@twoside \renewcommand{\noteraggedleft}{\RaggedRight} \renewcommand{\noteraggedright}{\RaggedRight} \else \renewcommand{\noteragged}{\RaggedRight} \fi}{% else if \ifthenelse{\equal{\exesheet@noteragged}{center}}{\if@twoside \renewcommand{\noteraggedleft}{\Centering} \renewcommand{\noteraggedright}{\Centering} \else \renewcommand{\noteragged}{\Centering} \fi}{% else if \end{verbatim}
The scale control option relies on calculations with *lengths*, which need to have a *global* scope.

For questions, assigned points will be added in \texttt{\sum@pts}, while for exercises, points accumulate in \texttt{\sum@exe}. These lengths are compared against \texttt{\exe@total} and \texttt{\sheet@total}. The \texttt{\exe@check} macro validates the calculations of the previous exercise when triggered by \texttt{\points}, \texttt{\totalexe} or \texttt{\totalpoints} macros. Percent symbols at end of lines are necessary to prevent unwanted spaces. \texttt{\exe@check} is also invoked within \texttt{\exs@process@checkpts} at the document’s end for a final check on the last exercise.

\begin{verbatim}
\newlength{\sheet@total}
\newlength{\sum@exe}
\newlength{\exe@total}
\newlength{\sum@pts}
\def\exe@currentlabel{none}
\newboolean{scale@valid}
\def\exe@check{
  \ifthenelse{\lengthtest{\sum@pts = 0pt}}{}{%
    \PackageWarningNoLine{exesheet}{Invalid option \texttt{noteragged=twoside} when the document \MessageBreak is not in two-side mode} }
  \PackageWarning{exesheet}{\texttt{noteragged} is not supported by the \texttt{noteragged} option} 
}

\exs@process@checkpts
\end{verbatim}
\def\exs@process@checkpts{
    \ifexesheet@checkpts
        \ifthenelse{\lengthtest{\sheet@total = 0pt}}{
            \PackageWarningNoLine{exesheet}{Option checkpts is true,}
            \PackageWarningNoLine{exesheet}{but \string\totalsheet\space is missing}
            \PackageWarningNoLine{exesheet}{in the preamble. \MessageBreak}
            \PackageWarningNoLine{exesheet}{See documentation}
        }{}
        \global\sum@exe=0pt
        \global\exe@total=0pt
        \global\sum@pts=0pt
        \setboolean{scale@valid}{true}
    \AtEndDocument{% final checking (global)
        \ifthenelse{\equal{\exe@currentlabel}{none}}{
            \ifthenelse{\lengthtest{\sum@pts = 0pt}}{
                \PackageWarningNoLine{exesheet}{checkpts: No points displayed}
            }{
                \ifthenelse{\lengthtest{\sheet@total = \sum@pts}}{
                    \PackageWarningNoLine{exesheet}{Total:
                        Sum of points \the\sheet@total\space is valid}
                }{
                    \PackageWarningNoLine{exesheet}{Total:
                        Sum of points is \the\sum@pts\space
                        instead of \the\sheet@total}
                }
            }
        }{% last exercise and final checking
            \exe@check
            \ifthenelse{\lengthtest{\sum@exe = 0pt}}{
                \PackageWarningNoLine{exesheet}{checkpts: No points displayed}
            }{
                \ifthenelse{\lengthtest{\sheet@total = \sum@exe}}{
                    \PackageWarningNoLine{exesheet}{Total:
                        Sum of points \the\sheet@total\space is valid}
                }{
                    \PackageWarningNoLine{exesheet}{Total:
                        Sum of points is \the\sum@exe\space
                        instead of \the\sheet@total}
                    \setboolean{scale@valid}{false}
                }
            }
            \ifthenelse{\boolean{scale@valid}}{
                \PackageWarningNoLine{exesheet}{Marking scheme checked without errors}
            }{
                \PackageWarningNoLine{exesheet}{Marking scheme checked with ERRORS! See above}
            }
        }
    }
}
\fi
7.7 Marking scheme commands

The \check@points macro, used by \points and \totalexe, triggers the marking scheme control (with \exe@check defined above) and sets label and lengths for the next exercise.

\newcommand*{\check@points}[1]{%
  \ifexesheet@checkpts%
    \exe@check% checks the previous exercise
    \gdef\exe@currentlabel{\exe@label}% for the upcoming exercise
    \global\sum@pts=0pt%
    \global\exe@total=#1pt%
    \global\advance\sum@exe by #1pt%
  \fi%
}%
\points

\definecolor{pointscolor}{named}{red}
\newcommand{\pointsstyle}{%
  \small\mdseries\sffamily\color{pointscolor}\fbox
}%
\newcommand*{\points}[1]{%
  \ifthenelse{\boolean{exesheet@questions}}{\hfill\pointsstyle{#1~%
    \ifthenelse{\lengthtest{#1pt < 2pt}}{\pointname}{\pointsname}}}%
    \check@points{#1}\
}%

To prevent spaces between the \fbox and its inner text, percent symbols are necessary. The test #1 < 2 doesn’t work with decimal numbers without \lengthtest, but it works with lengths.

\pts

\definecolor{ptscolor}{named}{red}
\newcommand{\ptsstyle}[1]{%
  \footnotesize\centering\sffamily\color{ptscolor}\(#1\)}
\newcommand*{\ptsmark}[1]{%
  \ifthenelse{\lengthtest{#1pt < 2pt}}{#1 \ptname}{#1 \ptsname}}
\newcommand*{\pts}[1]{%
  \ifexesheet@pts%
    \mbox{}\marginpar{\hspace{0pt}\ptsstyle{\ptsmark{#1}}}\ifexesheet@checkpts%
      \global\advance\sum@pts by #1pt%
  \fi%
}%
\ignorespaces
In the subsequent macros that utilize \marginpar, the presence of percent symbols and \ignorespaces is essential to prevent the occurrence of expanded blank spaces in the text (or the margin), where these macros are incorporated.

\definecolor{markingcolor}{named}{red}
\newcommand{\markingstyle}[1]{\footnotesize\sffamily\centering\color{markingcolor}\textbf{#1}}
% inner arguments enable the implementation of boxed styles
\newlength{ptsboxlength}
\setlength{ptsboxlength}{3.1em}
\cornersize{1}
\newcommand*{\totalexe}[1]{%
  \ifexesheet@pts%
    \mbox{}
    \marginpar{\hspace{0pt}\markingstyle{\ovalbox{\makebox[\ptsboxlength]{\ptsmark{#1}}}}}%
    \check@points{#1}%
  \fi%
  \ignorespaces%
}

\totalsheet
\newcommand*{\totalsheet}[1][]{%
  \global\sheet@total=#1pt
}

\note
The booleans \exesheet@pts and \exesheet@notes control the display of marginal notes. If \exesheet@pts is set to false, \exesheet@notes will be ignored. \noindent is required when using \justifying from the ragged2e package [7]. Within the \note@marginpar macro, enclosing \markingstyle in double braces helps prevent unintended formatting within the mandatory argument of \note. A vicious error occurs when using an \if ... \fi structure instead of \ifthenelse inside \note@marginpar (but only if \@twoside is true).

\definecolor{notecolor}{rgb}{0.0, 0.4, 0.0} % kind of dark green
\newcommand{\notestyle}[1]{\footnotesize\sffamily\color{notecolor} #1}
\newcommand{\note@marginpar}[1]{%
  \if@twoside%
    \marginpar[\noteraggedleft \#1]{\noteraggedright \#1}%
  \else%
    \marginpar[\noteragged \#1]%
  \fi%
}
\newcommand{\@note}[2][{}]{%
  \ifexesheet@pts%
    \mbox{}
    \note@marginpar{%
      \ifthenelse{\equal{#1}{}}{}{{%\noindent\hspace{0pt}\markingstyle{#1}}}%
      \ifthenelse{\boolean{exesheet@notes}}{\noindent\hspace{0pt}\notestyle #2}{}%    }
  \else%
    \@note[{}]
  \fi%
}

36
References

[1] The geometry package, Hideo Umeki, CTAN, v5.9 2020/01/02.


[6] translations – Internationalization of B\TeX\!\!\!\!\!\! Packages, Clemens Niederberger, CTAN, v1.12 2022/02/05.


[8] mparhack.sty, Tom Sgouros, Stefan Ulrich, CTAN, v1.5 2021/05/02.


[12] xsim – Exercise Sheets IMPROVED – the official successor of the exsheets package, Clemens Niederberger, CTAN, v0.21 2022/02/12.
[18] The exercisepoints Package, Henning Kerstan, CTAN, v1.2.3 2019/01/03.
[21] The eqexam Package – part of the Acro\TeX\ eDucation Bundle, D. P. Story, CTAN, 5.2 2021/02/26.
[22] cesenaexam — class file to typeset exams, Alex Pacini, CTAN, v2.0 2017/08/03.