The exesheet class and package

Antoine Missier
`antoine.missier@ac-toulouse.fr`

2023/10/25, v2.3

―――――――

Contents

1 Introduction

The exesheet *package* is designed for typesetting exercise or exam sheets. Additionally, the exesheet *class* loads the 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 schooldocs package for more details. The exesheet *class* is build upon the article class and forwards any unknown options to it.

There are many other packages dedicated to exercise sheets. Most of them suggest encapsulating each exercise within an environment. In contrast, exesheet starts each exercise with `\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 exesheet package is its specific settings for enumeration lists, which are suitable for numbering questions or answers within an exercise.

While other packages often offer more or less complex mechanisms for managing the placement of answers, exesheet doesn't aspire to such complexity. However, for all exercises within the sheet, you have the capability to display only questions, only 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.

Finally this package enables to display a detailed marking scheme in the margin, with optional explanations or remarks, and offering consistency control.

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

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

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

In the current document, a frame is utilized to emphasize output examples.

2 Titles

2.1 The `\exercise` command

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

Exercise 1 (to begin)

Try this first command; easy!

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)]`:

Exercise 2*** (difficult)

Calculate $1 + 1$.

`\exercisename`   The term "Exercise" is automatically translated into various languages[1] depending on the language that is loaded (via babel or polyglossia). You can redefine it using `\renewcommand`. A better approach is to use macros from the translations package by Clemens Niederberger (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:
`\renewcommand{\labelexercise}{\exercisename~\theexercise.}`

`\theexercise`   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:
`\renewcommand{\labelexercisestyle}{\rmfamily\color{black}}`[2].

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

Problem (Fermat's theorem)

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.

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.

Exercise 3

Part A (preliminary)

To begin, prepare your cup of tea.

Part B

Now you are ready to proceed with the current exercise.

---

[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.

The following macros manage formatting 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.

`\subpart*`   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.

`\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   Exercise 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.

4

\exname
\exlabel
\exsepmark
The abbreviation **Ex** can be modified by redefining `\exname` or with macros from the `translations` package. The `\exlabel` macro combines `\exname` with a period then the exercise number (given by the `exercise` counter), while `\exsepmark` typesets a long dash. You can alter these characteristics by redefining these commands.

`\exe*` The starred version doesn't display a separator, as demonstrated below:

Ex. 6 Another short exercise without a separator.

---

3 Enumerations and lists

3.1 List settings

`enumerate`
`\item`
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 `\setlist` command, a feature from the `enumitem` package by Javier Bezos.

Exercise 7

1. First question

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

2. Second question

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

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

`[setlist=⟨bool⟩]` The package option `setlist=false` prevents changes to enumeration and itemize lists and reverts to the default LaTeX settings (the default value is `true`).

3.2 List of exercises : the `exenumerate` environment

`exenumerate` 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 `\labelenumi` or `\labelenumii` commands.

[4]However, the `french` option of the `babel` package changes the behavior of `itemize` lists and employs long dashes as labels for each list level. This behavior can cause issues when mathematical content follows the dash symbol, as it might be mistaken for the minus sign. Thus, with the option `setlist=true`, the default LaTeX `itemize` list style is reinstated. The `\frenchsetup` command (from the `babel-french` package) can achieve several other settings.

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`, `tablenuma`, `tablitem`

`tablenum1`
`tablenuma`
`tablitem`
These three environments are employed to typeset brief questions (`tablenum1`), sub-questions (`tablenuma`) 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. The usage of the optional argument ⟨opt⟩ is explained in the documentation of this package. For example, similar to the `enumitem` package, `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.

Exercise 8

Calculate the derivative of the following functions:

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

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

For `tablenuma`, labels are letters, a, b, c, ..., enclosed in parentheses.

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

`\labelenumone`
`\labelenuma`

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

`tablenuma*`
`tablitem*`

When you intend to utilize `tablenuma` (or `tablitem`) immediately after inserting the `\item` command within an `enumerate` environment, a vertical misplacement occurs, as demonstrated in the following example:

1. **(a)** One **(b)** Two **(c)** Three

To achieve proper vertical spacing in such cases, we offer the starred environments `tablenuma*` and `tablitem*`, with corrected alignment as shown below:

1. **(a)** One **(b)** Two **(c)** Three

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

3.4 Items aligned by column: `colsenum, colsitem`

`colsenum`

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

Exercise 9

Calculate the derivative of the following functions:

1. $f(x) = \dfrac{1 - x^2}{e^x + e^{-x}}$,   **3.** $h(x) = \displaystyle\int_0^1 e^{xy}\,\mathrm{d}y$,   **5.** $l(x) = \displaystyle\int_{\frac{1}{x}}^x \dfrac{1}{\ln t}\,\mathrm{d}t$.

2. $g(x) = \ln\left(\dfrac{1 - x}{1 + x^2}\right)$,   **4.** $k(x) = \displaystyle\sum_{i=1}^\infty \dfrac{1}{x^i}$,

---

`colsenum*`

We will observe that, on each line, items are not necessarily properly aligned, which can result in ungraceful effects. On the other hand, the `colsenum` environment doesn't attempt to align columns from the bottom by adjusting the vertical spacing between items. If we desire this alignment (which is the default behavior in multicol), we can use the `colsenum*` environment (with the same syntax as `colsenum`). Here's what we obtain with `colsenum*`:

> **Exercise 10**
>
> Calculate the derivative of the following functions:
>
> **1.** $f(x) = \dfrac{1 - x^2}{\mathrm{e}^x + \mathrm{e}^{-x}}$,   **3.** $h(x) = \displaystyle\int_0^1 \mathrm{e}^{xy}\,\mathrm{d}y$,   **5.** $l(x) = \displaystyle\int_{\frac{1}{x}}^x \dfrac{1}{\ln t}\,\mathrm{d}t$.
>
> **2.** $g(x) = \ln\left(\dfrac{1 - x}{1 + x^2}\right)$,   **4.** $k(x) = \displaystyle\sum_{i=1}^{\infty} \dfrac{1}{x^i}$,

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 `colsenum` is that the label selection is automatic, based on the list level (and the language), unlike `tablenum1` or `tablenuma`.

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

4 Questions and solutions

4.1 Environments `questions` and `answers`

`questions`
`answers` The package offers two environments, `questions` and `answers`, which allow you to optionally show or hide questions and answers within exercises.

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

`\correctionstyle`
`correctioncolor` In the default case where both questions and answers are displayed, the answers are typeset using the `\correctionstyle` style, which utilizes the color `correctioncolor`. You can modify this color using the `\definecolor` macro[5]. By default, `\definecolor{correctioncolor}{rgb}{0,0.2,0.6}` is used, resulting in a kind of dark blue.

`\correctionname` Furthermore, when using `output=both` the title **Correction** is displayed at the beginning of `answers` environments. This title is defined by the `\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 `\correctionstyle` will be applied to the title as well as the entire environment. Here's an example to illustrate this:

---

[5]The `\definecolor` command is provided by the xcolor package developed by Uwe Kern, which is automatically loaded by exesheet.

> **Exercise 11**
>
> **1.** Is the exesheet package useful ?
>
> **2.** Aren't there any other packages that deal with exercises ?
>
> **Correction**
>
> **1.** Yes, the exesheet package is indeed 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.

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. Moreover, the versions package 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.

`answers[⟨level⟩]`   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.

`answers*`   The starred version `answers*` doesn't display the **Correction** title.

4.3 Commands `\question`, `\answer` and `\answerspace`

`\question`
`\answer`
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*`
`\answer*`
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}`.

`\answerspace`
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{⟨height⟩}` macro, in which the parameter ⟨*height*⟩ is a valid length, e.g. `\answerspace{3cm}`.

`[answerspace=⟨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 enables the display of a detailed marking scheme in the margins, along with optional comments and explanations about answers.

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}]`:

Exercise 12   5 points

Try to read this document to the end without drinking tea and you get five points.

When only the correction is displayed in an exercise, the `\points` macro doesn't show the points. Printing answers along with the point scale will be discussed in section 5.5, which includes another `\totalpoints` macro.

`\pointsname`
`\pointname`
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).

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

---

[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

`\pts` When exercises are typeset using the `\exe` macro or as a list with the `exenumerate` 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 ⟨num⟩ parameter represents the number of points assigned to the exercise. Here's an example with `\exe\pts{3}...` `\exe\pts{1.5}...`

(3 pts) **Ex. 13** — The first short exercise with a marking scheme.

(1.5 pt) **Ex. 14** — The second one.

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

`[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.

`[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 (which exesheet automatically includes for documents in two-side mode) and *running 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 `\totalexe{⟨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*{⟨num⟩}` 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 ⟨*comment*⟩ argument you can use `\\` to create a line break or even `\\[⟨height⟩]` to adjust the line spacing by ⟨*height*⟩.

`\note[⟨num⟩]`   Placing `\note*{⟨num⟩}\note{⟨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}`.

---

4 pts

Exercise 15

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

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 simplifying

1. We calculate:

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

TRUE.

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

Other method:
$\frac{1}{x\ln x} > 0$ on $[2,\mathrm{e}]$
whereas $-\ln 2 < 0$

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

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

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 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 $F'$\\1 for the sign of $F'$ and conclusion}`.

`markingcolor`
`\markingstyle`
`\ptsboxlength`

The color and style for displaying points in `\totalexe` and `\note*` can be customized using `markingcolor` and `\markingstyle`, respectively. The oval box produced by `\totalexe` is created using the `\ovalbox` command from the fancybox package (by Timothy Van Zandt), 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.

`notecolor`
`\notestyle`

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

`[display=⟨opt⟩]`

The `display` 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`, `\totalexe`, `\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.

`[marginpos=⟨opt⟩]`

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

`[marginwidth=⟨opt⟩]`

The margin layout is governed by the `marginwidth` 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). 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.

[noteragged=⟨*opt*⟩]   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 LaTeX'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

\totalpoints   The `\totalpoints{`⟨*num*⟩`}` macro serves as a replacement for `\points` when using a comprehensive marking scheme. When the scale isn't visible, it functions similarly to `\points`, and when the scale is shown, it's akin to `\totalexe`. For instance, in exercise 15, we could have used `\totalpoints` instead of `\totalexe`. In this case, when the detailed marking scheme is not displayed, the total points would be presented similarly to exercise 5.1 rather than being absent.

5.6   Marking scheme consistency checking

[checkpts=⟨*bool*⟩]   The marking scheme can be checked out[7] 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 `\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.

---

[7]Thanks to Denis Bitouzé for his suggestion about this feature.

6 Options

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.

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

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 You can also use the `\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 `\begin{document}`.

[correct=⟨*opt*⟩] 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 (deprecated) 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, the older commands are still supported for compatibility reasons and are outlined here. While these commands will trigger a warning message, they remain functional. 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
\displaypoints `display=pts`; `\displaynotes` means `display=notes`, and `\displaynotesright`
\displaynotes corresponds to `display=notes` and `marginpos=right`.
\displaynotesright

7 Implementation

7.1 Options and required packages

The exesheet class is build 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.

```
 1 ⟨*class⟩
 2 \RequirePackage{kvoptions}
 3 \DeclareBoolOption[true]{exetoc}
 4 \DeclareBoolOption[true]{setlist}
 5 \DeclareStringOption[both]{output}
 6 \DeclareStringOption[none]{display}
 7 \DeclareBoolOption[false]{answerspace}
 8 \DeclareStringOption[left]{marginpos}
 9 \DeclareStringOption[expand]{marginwidth}
10 \DeclareStringOption[left]{noteragged}
11 \DeclareBoolOption[false]{checkpts}
12 \DeclareStringOption[false]{correct}
13 \DeclareOption*{\PassOptionsToClass{\CurrentOption}{article}}
14 \ProcessOptions \relax
15 \LoadClass{article}
16 \RequirePackage{exesheet}
17 \RequirePackage{schooldocs}
18 ⟨/class⟩
```

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 and subsequently disabled (further clarification follows below).

```
19 ⟨*package⟩
20 \@ifclassloaded{exesheet}{}{
21     \RequirePackage{kvoptions}
22     \DeclareBoolOption[true]{exetoc}
23     \DeclareBoolOption[true]{setlist}
24     \DeclareStringOption[both]{output}
25     \DeclareStringOption[none]{display}
26     \DeclareBoolOption[false]{answerspace}
27     \DeclareStringOption[left]{marginpos}
28     \DeclareStringOption[expand]{marginwidth}
29     \DeclareStringOption[left]{noteragged}
30     \DeclareBoolOption[false]{checkpts}
31     \DeclareStringOption[false]{correct}
32 }
33
34 \ProcessKeyvalOptions*
35
36 \def\exs@process@dynoptions{
```

```
37        \exs@process@output
38        \exs@process@display
39        \exs@process@noteragged
40 } % answerspace do not need a special process macro
41
42 \AtEndOfPackage{\exs@process@dynoptions}
43 \AtBeginDocument{
44      \newif\ifexesheet@multicol
45      \@ifpackageloaded{multicol}{
46          \exesheet@multicoltrue}{\exesheet@multicolfalse}
47          % configuring the rule color within answers environments
48      \exs@process@setlist
49      \exs@process@marginpos
50      \exs@process@marginwidth
51      \exs@process@checkpts
52      \exs@process@correct
53      \DisableKeyvalOption[action=warning,package=exesheet]{exesheet}{setlist}
54      \DisableKeyvalOption[action=warning,package=exesheet]{exesheet}{marginpos}
55      \DisableKeyvalOption[action=warning,package=exesheet]{exesheet}{marginwidth}
56      \DisableKeyvalOption[action=warning,package=exesheet]{exesheet}{checkpts}
57      \DisableKeyvalOption[action=warning,package=exesheet]{exesheet}{correct}
58 }
59
```

\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.

```
60 \def\exesheetset#1{\setkeys{exesheet}{#1}\exs@process@dynoptions}
61
```

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 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)
is loaded exclusively for documents in twoside mode.

```
62 \RequirePackage{ifthen}
63 \@ifpackageloaded{geometry}{}{\RequirePackage{geometry}}
64 \RequirePackage{xcolor}
65 \RequirePackage[shortlabels]{enumitem}
66 \RequirePackage{tasks}[2020/08/19]
67 \RequirePackage{versions}
68 \RequirePackage{fancybox}
69 \RequirePackage{translations}
70 \RequirePackage{ragged2e}
71 \ifthenelse{\boolean{@twoside}}{\RequirePackage{mparhack}}{}
72
```

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. This package automatically detects the language being used, as loaded by babel or polyglossia.

Accented characters cannot be utilized here, as they might not be recognized if exesheet is loaded before any other package (typically when it is loaded as a class). As a workaround, we rely on basic LaTeX control sequences to generate them.

```
73 \DeclareTranslationFallback{exesheet-exercise}{Exercise}
74 \DeclareTranslationFallback{exesheet-subpart}{Part}
75 \DeclareTranslationFallback{exesheet-annex}{Annex}
76 \DeclareTranslationFallback{exesheet-ex}{Ex}
77 \DeclareTranslationFallback{exesheet-points}{points}
78 \DeclareTranslationFallback{exesheet-point}{point}
79 \DeclareTranslationFallback{exesheet-correction}{Correction}
80 \DeclareTranslationFallback{exesheet-pts}{pts}
81 \DeclareTranslationFallback{exesheet-pt}{pt}
82
83 \DeclareTranslation{english}{exesheet-exercise}{Exercise}
84 \DeclareTranslation{english}{exesheet-subpart}{Part}
85 \DeclareTranslation{english}{exesheet-annex}{Annex}
86 \DeclareTranslation{english}{exesheet-ex}{Ex}
87 \DeclareTranslation{english}{exesheet-points}{points}
88 \DeclareTranslation{english}{exesheet-point}{point}
89 \DeclareTranslation{english}{exesheet-correction}{Correction}
90 \DeclareTranslation{english}{exesheet-pts}{pts}
91 \DeclareTranslation{english}{exesheet-pt}{pt}
92
93 \DeclareTranslation{french}{exesheet-exercise}{Exercice}
94 \DeclareTranslation{french}{exesheet-subpart}{Partie}
95 \DeclareTranslation{french}{exesheet-annex}{Annexe}
96 \DeclareTranslation{french}{exesheet-ex}{Ex}
97 \DeclareTranslation{french}{exesheet-points}{points}
98 \DeclareTranslation{french}{exesheet-point}{point}
99 \DeclareTranslation{french}{exesheet-correction}{Correction}
100 \DeclareTranslation{french}{exesheet-pts}{pts}
101 \DeclareTranslation{french}{exesheet-pt}{pt}
102
103 \DeclareTranslation{german}{exesheet-exercise}{\"Ubung}
104 \DeclareTranslation{german}{exesheet-subpart}{Teil}
105 \DeclareTranslation{german}{exesheet-annex}{Anhang}
106 \DeclareTranslation{german}{exesheet-ex}{\"Ub}
107 \DeclareTranslation{german}{exesheet-points}{Punkte}
108 \DeclareTranslation{german}{exesheet-point}{Punkt}
109 \DeclareTranslation{german}{exesheet-correction}{Verbesserung}
110 \DeclareTranslation{german}{exesheet-pts}{P.}
111 \DeclareTranslation{german}{exesheet-pt}{P.}
112
113 \DeclareTranslation{spanish}{exesheet-exercise}{Ejercicio}
114 \DeclareTranslation{spanish}{exesheet-subpart}{Parte}
115 \DeclareTranslation{spanish}{exesheet-annex}{Anexo}
116 \DeclareTranslation{spanish}{exesheet-ex}{Ej}
117 \DeclareTranslation{spanish}{exesheet-points}{puntos}
118 \DeclareTranslation{spanish}{exesheet-point}{punto}
119 \DeclareTranslation{spanish}{exesheet-correction}{Correcci\'on}
120 \DeclareTranslation{spanish}{exesheet-pts}{ptos}
```

```
121 \DeclareTranslation{spanish}{exesheet-pt}{pto}
122
123 \DeclareTranslation{italian}{exesheet-exercise}{Esercizio}
124 \DeclareTranslation{italian}{exesheet-subpart}{Parte}
125 \DeclareTranslation{italian}{exesheet-annex}{Annesso}
126 \DeclareTranslation{italian}{exesheet-ex}{Es}
127 \DeclareTranslation{italian}{exesheet-points}{punti}
128 \DeclareTranslation{italian}{exesheet-point}{punto}
129 \DeclareTranslation{italian}{exesheet-correction}{Correzione}
130 \DeclareTranslation{italian}{exesheet-pts}{pti}
131 \DeclareTranslation{italian}{exesheet-pt}{pt}
132
133 \DeclareTranslation{portuges}{exesheet-exercise}{Exerc\'icio}
134 \DeclareTranslation{portuges}{exesheet-subpart}{Parte}
135 \DeclareTranslation{portuges}{exesheet-annex}{Anexo}
136 \DeclareTranslation{portuges}{exesheet-ex}{Ex}
137 \DeclareTranslation{portuges}{exesheet-points}{pontos}
138 \DeclareTranslation{portuges}{exesheet-point}{ponto}
139 \DeclareTranslation{portuges}{exesheet-correction}{Corre\c c\~ao}
140 \DeclareTranslation{portuges}{exesheet-pts}{pts}
141 \DeclareTranslation{portuges}{exesheet-pt}{pt}
142
143 \newcommand*\exercisename{\GetTranslation{exesheet-exercise}}
144 \newcommand*\subpartname{\GetTranslation{exesheet-subpart}}
145 \newcommand*\annexname{\GetTranslation{exesheet-annex}}
146 \newcommand*\exname{\GetTranslation{exesheet-ex}}
147 \newcommand*\pointsname{\GetTranslation{exesheet-points}}
148 \newcommand*\pointname{\GetTranslation{exesheet-point}}
149 \newcommand*\correctionname{\GetTranslation{exesheet-correction}}
150 \newcommand*\ptsname{\GetTranslation{exesheet-pts}}
151 \newcommand*\ptname{\GetTranslation{exesheet-pt}}
152
```

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}`.

`\exercise`

```
153 \newcounter{exercise}
154 \newcounter{exe@check}
155
156 \newcommand{\labelexercise}{\exercisename\space \theexercise}
157 \newcommand{\labelexercisestyle}{}
158 \newcommand*{\@exercise}[1][]{%
159     \refstepcounter{exercise}
160     \subsection*{\labelexercisestyle\labelexercise\enskip #1}
161     \ifexesheet@exetoc
162         \addcontentsline{toc}{subsection}{\labelexercise}
163     \fi
164     \ifexesheet@checkpts
165      \setcounter{exe@check}{\value{exercise}}
166         \def\exe@label{\exercisename\space\theexe@check}
167     \fi
168 }
169 \newcommand*{\@@exercise}[2][]{%
170     \subsection*{\labelexercisestyle #2\enskip #1}
171     \setcounter{subpart}{0} % resets the parts counter
172     \ifexesheet@exetoc
173         \addcontentsline{toc}{subsection}{#2}
174     \fi
175     \ifexesheet@checkpts \def\exe@label{#2} \fi
176 }
177 \newcommand{\exercise}{\@ifstar{\@@exercise}{\@exercise}}
178
```

`\subpart`

```
179 \newcounter{subpart}[exercise] %
180 \renewcommand{\thesubpart}{\Alph{subpart}}
181
182 \newcommand{\labelsubpart}{\subpartname~\thesubpart}
183 \newcommand{\labelsubpartstyle}{}
184 \newcommand*{\@subpart}[1][]{%
185     \refstepcounter{subpart}%
186     \subsubsection*{\labelsubpartstyle\labelsubpart\enskip #1}
187     \ifexesheet@exetoc
188         \addcontentsline{toc}{subsubsection}{\labelsubpart}
189     \fi
190 }
191 \newcommand*{\@@subpart}[2][]{%
192     \subsubsection*{\labelsubpartstyle #2\enskip #1}
193     \ifexesheet@exetoc
194         \addcontentsline{toc}{subsubsection}{#2}
195     \fi
196 }
197 \newcommand{\subpart}{\@ifstar{\@@subpart}{\@subpart}}
198
```

`\annex`

```
199 \newcommand{\annexstyle}{\MakeUppercase}
200 \newcommand*{\annex}[1][]{%
201     \subsection*{\mbox{}\hfill\annexstyle{\annexname} #1\hfill\mbox{}}
```

```
202        \ifexesheet@exetoc
203            \addcontentsline{toc}{subsection}{\annexname}
204        \fi
205 }
206
```

```
207 \newcommand{\exlabel}{\exname.~\theexercise}
208 \newcommand{\exsepmark}{---}
209 \newcommand{\@exe}{\bigskip\refstepcounter{exercise}
210        \ifexesheet@checkpts
211            \setcounter{exe@check}{\value{exercise}}
212            \def\exe@label{\exname\space\theexe@check}
213        \fi
214        \par\noindent\textbf{\exlabel~\exsepmark}~}
215 \newcommand{\@@exe}{\bigskip\refstepcounter{exercise}
216        \ifexesheet@checkpts
217            \setcounter{exe@check}{\value{exercise}}
218            \def\exe@label{\exname\space\theexe@check}
219        \fi
220        \par\noindent\textbf{\exlabel}~}
221 \newcommand{\exe}{\@ifstar{\@@exe}{\@exe}}
222
```

7.4   Enumerations and lists

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

```
223 \newenvironment{exenumerate}[1][]{%
224        \setlist[enumerate]{font=\bfseries}
225        \setlist[enumerate,1]{leftmargin=1.5em,
226            itemsep=3ex plus 1ex minus 1ex,topsep=3ex plus 1ex minus 1ex}
227        \setlist[enumerate,3]{noitemsep,nolistsep}
228        \setlist[itemize]{noitemsep,nolistsep}
229        \begin{enumerate}[#1]
230            }{\end{enumerate}}
231
```

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, depending on whether exesheet is loaded
before or after babel.

```
232 \newcommand\standardfrenchlists{%
233        \@ifpackagewith{babel}{french}{
234            \frenchsetup{StandardLists=true}
235        }{}
236 }
237 \ifexesheet@setlist
238        \standardfrenchlists
239        % must be executed here (and not at begin doc) if loaded after babel
```

```
240 \fi
241
242 \newcommand\labelenumone{\arabic{task}.}
243 \newcommand\labelenuma{(\alph{task})}
244
245 \def\exs@process@setlist{% must be executed at begin document
246   \ifexesheet@setlist
247     \standardfrenchlists % executed at begin doc if loaded before babel
248     \setlist[enumerate]{font=\bfseries}
249     \setlist[enumerate,1]{topsep=1.5ex plus 1ex minus 1ex,leftmargin=1.5em}
250   \fi
```

tablenum1   The \NewTasks command is part of the tasks package. It enables the definition
tablenuma   of the environments tablenum1, tablenuma and tablitem. Horizontal spacing is
            adjusted to ensure proper alignment with items in other enumerate (or itemize)
            environments.

```
251   \ifexesheet@setlist
252     \settasks{label-format=\bfseries}
253     \NewTasksEnvironment[label=\labelenumone,
254         column-sep=1em,label-align=right,
255         item-indent=1.5em,label-width=1em,label-offset=0.5em,
256         after-item-skip=0.5ex plus 0.5ex minus 0.5ex]{tablenum1}[\item](2)
257     \NewTasksEnvironment[label=\labelenuma,ref=\alph{task},
258         column-sep=1em,label-align=right,
259         item-indent=2.15em,label-width=1.6em,label-offset=0.5em,
260         after-item-skip=0.5ex plus 0.5ex minus 0.5ex]{tablenuma}[\item](2)
261   \else
262     \NewTasksEnvironment[label=\labelenumone,
263         column-sep=1em,label-align=right,
264         label-width=1em,label-offset=0.5em,
265         after-item-skip=0.5ex plus 0.5ex minus 0.5ex]{tablenum1}[\item](2)
266     \NewTasksEnvironment[label=\labelenuma,ref=\alph{task},
267         column-sep=1em,label-align=right,
268         item-indent=2.15em,label-width=1.6em,label-offset=0.5em,
269         after-item-skip=0.5ex plus 0.5ex minus 0.5ex]{tablenuma}[\item](2)
270   \fi
271 } % end of macro \exs@process@setlist
272
```

tablitem

```
273 \NewTasksEnvironment[label=\labelitemi,
274     label-align=right,
275     item-indent=2.5em,label-offset=0.5em,
276     after-item-skip=0.5ex plus 0.5ex minus 0.5ex]{tablitem}[\item](2)
277
```

tablenuma*   The starred environments tablenuma* and tablitem* are designed to be employed
tablitem*   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.

```
278 \newenvironment{tablenuma*}{%
279     \mbox{}\vspace{-1.667\baselineskip}\begin{tablenuma}}{
280     \end{tablenuma}}
281 \newenvironment{tablitem*}{%
282     \mbox{}\vspace{-1.667\baselineskip}\begin{tablitem}}{
```

```
283        \end{tablitem}}
284
```

colsenum

```
285 \newenvironment{colsenum}[2][]{%
286        \setlength{\multicolsep}{2ex}
287        \raggedcolumns % default is \flushcolumns
288        \begin{multicols}{#2} % #2 = number of columns
289        \begin{enumerate}[#1] % #1 = options of enumerate
290        }{
291        \end{enumerate}
292        \end{multicols}
293 }
294
```

colsenum*

```
295 \newenvironment{colsenum*}[2][]{%
296        \setlength{\multicolsep}{2ex}
297        \begin{multicols}{#2} % #2 = number of columns
298        \begin{enumerate}[#1] % #1 = options of enumerate
299        }{
300        \end{enumerate}
301        \end{multicols}
302 }
303
```

colsitem

```
304 \newenvironment{colsitem}[2][]{%
305        \setlength{\multicolsep}{2ex}
306        \raggedcolumns
307        \begin{multicols}{#2}
308        \begin{itemize}[#1]
309        }{
310        \end{itemize}
311        \end{multicols}
312 }
313
```

colsitem*

```
314 \newenvironment{colsitem*}[2][]{%
315        \setlength{\multicolsep}{2ex}
316        \begin{multicols}{#2}
317        \begin{itemize}[#1]
318        }{
319        \end{itemize}
320        \end{multicols}
321 }
322
```

7.5   Questions and answers

\exs@process@output   The booleans `exesheet@questions` and `exesheet@answers` governs the visibility
of their corresponding environments. These booleans are configured through the
`output` key option within the `\exs@process@output` macro.

```
323 \newboolean{exesheet@questions}\setboolean{exesheet@questions}{true}
```

```
324 \newboolean{exesheet@answers}\setboolean{exesheet@answers}{true}
325
326 \def\exs@process@output{
327     \ifthenelse{\equal{\exesheet@output}{questions}}{
328         \setboolean{exesheet@questions}{true}
329         \setboolean{exesheet@answers}{false}
330     }{% else if
331     \ifthenelse{\equal{\exesheet@output}{answers}}{
332         \setboolean{exesheet@questions}{false}
333         \setboolean{exesheet@answers}{true}
334         \exesheet@answerspacefalse
335     }{% else if
336     \ifthenelse{\equal{\exesheet@output}{both}}{
337         \setboolean{exesheet@questions}{true}
338         \setboolean{exesheet@answers}{true}
339         \exesheet@answerspacefalse
340     }{% else
341     \PackageWarning{exesheet}{Value '\exesheet@output'
342         is not supported by 'output' option}
343     }}}
344 }
345
```

questions   We utilize the versions package developed by Uwe Lück, which introduces the macros `\comment` and `\endcomment`. These macros facilitate conditional displays, a technique also employed in the verbatim and version packages. Additionally, the notable codesection package offers the capability to enclose optional code between `\BeginCodeSection{⟨skip⟩}` and `\EndCodeSection{⟨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 `\comment` and `\endcomment`. Several of our tests use the LaTeX syntax `\ifthenelse{\bolean{...}}` since `\comment` and `\endcomment` can sometimes interfere with the TeX structure `\if ...\else ...\fi`.

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

```
346 \newcounter{exe@ini}
347 \newcounter{subpart@ini}
348
349 \newenvironment{questions}{
350     \ifthenelse{\boolean{exesheet@questions}}{%
351         \setcounter{exe@ini}{\value{exercise}}
352         \setcounter{subpart@ini}{\value{subpart}}
353     }{\comment}
354 }{\ifthenelse{\boolean{exesheet@questions}}{}{\endcomment}}
355
```

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

```
356 \newcounter{@toclevel}
357 \newcommand{\set@toclevel}[1][]{
358     \ifthenelse{\equal{#1}{}}{
359         \ifthenelse{\value{exercise} > \value{exe@ini}}{
360             \setcounter{@toclevel}{1}
361         }{% else
362         \ifthenelse{\equal{\the\@enumdepth}{0}}{
363             % we're not in an enumerate environment
364             \ifthenelse{\(\value{subpart} > \value{subpart@ini}\)
365                 \or \(\value{subpart} = 0\)}{
366                 \setcounter{@toclevel}{2}
367             }{\setcounter{@toclevel}{3}}
368         }{\setcounter{@toclevel}{4}}}
369     }{\setcounter{@toclevel}{#1}}}
370
```

The internal macro `\typeset@correctionname`, displays the term "Correction" at the appropriate level.

```
371 \definecolor{correctioncolor}{rgb}{0,0.2,0.6} % kind of dark blue
372 \newcommand{\correctionstyle}{\color{correctioncolor}}
373
374 \newcommand{\typeset@correctionname}{
375     \ifthenelse{\value{@toclevel} = 1}{
376         \section*{\correctionstyle\correctionname}
377         \ifexesheet@exetoc
378             \addcontentsline{toc}{section}{\correctionname}
379         \fi
380         \setcounter{exercise}{0}
381     }{% else if
382     \ifthenelse{\value{@toclevel} = 2}{%
383         \subsection*{\correctionstyle\correctionname}
384         \ifexesheet@exetoc
385             \addcontentsline{toc}{subsection}{\correctionname}
386         \fi
387         \setcounter{subpart}{0}
388     }{% else if
389     \ifthenelse{\value{@toclevel} = 3}{%
390         \subsubsection*{\correctionstyle\correctionname}
391         \ifexesheet@exetoc
392             \addcontentsline{toc}{subsubsection}{\correctionname}
393         \fi
394     }{% else
395     \par\textbf{\correctionstyle\correctionname}\par
396     }}}
397 }
398
```

Then we proceed to define the `answers` environment.

```
399 \newenvironment{answers}[1][]{% #1 is the optional level
400     \ifthenelse{\boolean{exesheet@answers}}{%
401         \ifthenelse{\boolean{exesheet@questions}}{
402             \set@toclevel[#1]
403             \typeset@correctionname
404             \correctionstyle%
405             \ifexesheet@multicol
```

```
406                    \renewcommand{\columnseprulecolor}{\color{correctioncolor}}
407            \fi
408        }{}
409    }{\comment}
410 }{\ifthenelse{\boolean{exesheet@answers}}{}{\endcomment}}
411
412 \newenvironment{answers*}{
413    \ifthenelse{\boolean{exesheet@answers}}{\correctionstyle}{\comment}
414 }{\ifthenelse{\boolean{exesheet@answers}}{}{\endcomment}}
415
```

In the `answers` environment, when placing `\correctionstyle` before `\subsubsection` (as in the case of `\typeset@correctionname`), the preceding vertical space may become too wide.

```
416 \newcommand{\@question}[1]{\ifexesheet@questions #1\fi}
417 \newcommand{\@@question}[1]{%
418    \ifexesheet@questions\ifexesheet@answers \else #1\fi\fi}
419 \newcommand{\question}{\@ifstar{\@@question}{\@question}}
420
```

```
421 \newcommand{\@answer}[1]{%
422    \ifexesheet@answers%
423        \ifexesheet@questions {\correctionstyle #1}\else #1\fi
424    \fi
425 }
426 \newcommand{\@@answer}[1]{%
427    \ifexesheet@answers\ifexesheet@questions \else #1\fi\fi}
428 \newcommand{\answer}{\@ifstar{\@@answer}{\@answer}}
429
```

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

```
430 \newcommand\answerspace[1]{
431    \ifexesheet@answerspace \par\vspace{#1} \fi}
432
```

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`.

```
433 \newboolean{exesheet@pts}
434 \newboolean{exesheet@notes}
435
436 \def\exs@process@display{
```

```
437        \ifthenelse{\equal{\exesheet@display}{pts}}{
438            \setboolean{exesheet@pts}{true}
439            \setboolean{exesheet@notes}{false}
440        }{% else if
441        \ifthenelse{\equal{\exesheet@display}{notes}}{
442            \setboolean{exesheet@pts}{true}
443            \setboolean{exesheet@notes}{true}
444        }{% else if
445        \ifthenelse{\equal{\exesheet@display}{none}}{
446            \setboolean{exesheet@pts}{false}
447            \setboolean{exesheet@notes}{false}
448        }{% else
449        \PackageWarning{exesheet}{Value `\exesheet@display'
450            is not supported by `display' option}
451        }}}
452 }
453
```

\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).

```
454 \newboolean{exesheet@leftmargin}
455
456 \def\exs@process@marginpos{
457        \ifthenelse{\equal{\exesheet@marginpos}{left}}{
458            \if@twoside%
459                \PackageWarningNoLine{exesheet}{The default `marginpos'
460                    option \MessageBreak
461                    for two-sided documents is `outer'.\MessageBreak
462                    To change the side, use `inner'}
463                \def\exesheet@marginpos{outer}
464                \setboolean{exesheet@leftmargin}{false}
465                \normalmarginpar
466            \else% default
467                \setboolean{exesheet@leftmargin}{true}
468                \reversemarginpar
469            \fi
470        }{% else if
471        \ifthenelse{\equal{\exesheet@marginpos}{right}}{
472            \if@twoside%
473                \PackageWarningNoLine{exesheet}{The default `marginpos'
474                    option \MessageBreak
475                    for two-sided documents is `outer'.\MessageBreak
476                    To change the side, use `inner'}
477                \def\exesheet@marginpos{outer}
478            \fi
479            \setboolean{exesheet@leftmargin}{false}
480            \normalmarginpar
481        }{% else if
482        \ifthenelse{\equal{\exesheet@marginpos}{inner}}{
483            \setboolean{exesheet@leftmargin}{true}
484            \reversemarginpar
485        }{% else if
```

```
486    \ifthenelse{\equal{\exesheet@marginpos}{outer}}{
487        \setboolean{exesheet@leftmargin}{false}
488        \normalmarginpar
489    }{% else
490    \PackageWarningNoLine{exesheet}{The value '\exesheet@marginpos'
491        is not supported by the 'marginpos' option}
492    }}}}
493 }
494
```

$\exs@process@marginwidth$ The `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)[8].

When `display=notes`, the additional length of `1 in` corresponds to the default free space to the left of `\oddsidemargin`.

The macros `\standardmarginwidthfactor` and `\largemarginwidthfactor` represent the ratios between the total margin width and `\marginparwidth`.

```
495 \def\standardmarginwidthfactor{0.6}
496 \def\largemarginwidthfactor{0.8}
497
498 \newcommand*{\leftnotemarginwidth}[1]{
499    \setlength{\marginparwidth}{\oddsidemargin}
500    \addtolength{\marginparwidth}{1in}
501    \addtolength{\marginparwidth}{-\marginparsep}
502    \setlength{\marginparwidth}{#1\marginparwidth}
503 }
504
505 \newcommand*\rightnotemarginwidth[1]{
506    \setlength{\marginparwidth}{\paperwidth}
507    \addtolength{\marginparwidth}{-\textwidth}
508    \addtolength{\marginparwidth}{-\oddsidemargin}
509    \addtolength{\marginparwidth}{-\marginparsep}
510    \addtolength{\marginparwidth}{-1in}
511    \setlength{\marginparwidth}{#1\marginparwidth}
512 }
513
514 \def\exesheet@smallmargins{
515    \geometry{hmarginratio=1:1}
516    \leftnotemarginwidth{\standardmarginwidthfactor}
517 }
518 \def\exesheet@standardmargins{
519    \ifexesheet@leftmargin
520        \geometry{hmarginratio=3:2}
521        \leftnotemarginwidth{\standardmarginwidthfactor}
522    \else
523        \geometry{hmarginratio=2:3}
524        \rightnotemarginwidth{\standardmarginwidthfactor}
525    \fi
526 }
527 \def\exesheet@largemargins{
528    \ifexesheet@leftmargin
529        \geometry{hmarginratio=3:1}
```

---

[8]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.

```
530         \leftnotemarginwidth{\largemarginwidthfactor}
531     \else
532         \geometry{hmarginratio=1:3}
533         \rightnotemarginwidth{\largemarginwidthfactor}
534     \fi
535 }
536
537 \def\exs@process@marginwidth{
538     \ifthenelse{\equal{\exesheet@marginwidth}{standard}}{
539         \ifthenelse{\equal{\exesheet@display}{none}}{
540             \if@twoside
541                 \exesheet@standardmargins
542             \else
543                 \exesheet@smallmargins
544             \fi
545         }{% else display=pts or display=notes
546             \exesheet@standardmargins
547         }
548     }{% else if
549     \ifthenelse{\equal{\exesheet@marginwidth}{expand}}{
550         \ifthenelse{\equal{\exesheet@display}{none}}{
551             \if@twoside
552                 \exesheet@standardmargins
553             \else
554                 \exesheet@smallmargins
555             \fi
556         }{% else if
557         \ifthenelse{\equal{\exesheet@display}{pts}}{
558             \exesheet@standardmargins
559         }{% else display=notes
560             \exesheet@largemargins
561         }}
562     }{% else if
563         \ifthenelse{\equal{\exesheet@marginwidth}{unset}}{
564         % do nothing
565     }{% else
566     \PackageWarningNoLine{exesheet}{The value '\exesheet@marginwidth'
567          is not supported by the 'marginwidth' option}
568     }}}
569 }
570
```

For a two-sided document, the geometry package does not correctly set the width of the margin paragraph by default; it's too wide. Therefore, we provide an explicit setting here, which is useful when marginwidth=unset. Otherwise, the setting is handled by the marginwidth key option.

```
571 \if@twoside \rightnotemarginwidth{0.5} \fi
572
```

\exs@process@noteragged   The noteragged option can take one of the following values: left, right, center, justify or twoside. When working with a two-sided document, \marginpar can be used with an optional parameter to distinguish left from right contents. In this context, we employ \noteraggedleft and \noteraggedright instead of

\noteragged. The ragged2e package by Martin Schröder offers the commands \RaggedLeft, \RaggedRight, \Centering, and \justifying. These commands yield better results compared to the standard \raggedleft, \raggedright and \centering commands. Margin paragraphs are justified by default in LaTeX.

```
573 \newcommand{\noteragged}{}
574 \newcommand{\noteraggedleft}{}
575 \newcommand{\noteraggedright}{}
576
577 \def\exs@process@noteragged{
578     \ifthenelse{\equal{\exesheet@noteragged}{left}}{
579         \if@twoside
580             \renewcommand{\noteraggedleft}{\RaggedLeft}
581             \renewcommand{\noteraggedright}{\RaggedLeft}
582         \else
583             \renewcommand{\noteragged}{\RaggedLeft}
584         \fi
585     }{% else if
586     \ifthenelse{\equal{\exesheet@noteragged}{right}}{
587         \if@twoside
588             \renewcommand{\noteraggedleft}{\RaggedRight}
589             \renewcommand{\noteraggedright}{\RaggedRight}
590         \else
591             \renewcommand{\noteragged}{\RaggedRight}
592         \fi
593     }{% else if
594     \ifthenelse{\equal{\exesheet@noteragged}{center}}{
595         \if@twoside
596             \renewcommand{\noteraggedleft}{\Centering}
597             \renewcommand{\noteraggedright}{\Centering}
598         \else
599             \renewcommand{\noteragged}{\Centering}
600         \fi
601     }{% else if
602     \ifthenelse{\equal{\exesheet@noteragged}{justify}}{
603         \renewcommand{\noteraggedleft}{\justifying} % equiv to nothing
604         \renewcommand{\noteraggedright}{\justifying}
605         \renewcommand{\noteragged}{\justifying}
606     % justify is the default LaTeX setting
607     }{% else  if
608     \ifthenelse{\equal{\exesheet@noteragged}{twoside}}{
609         \if@twoside
610             \renewcommand{\noteraggedleft}{\RaggedLeft}
611             \renewcommand{\noteraggedright}{\RaggedRight}
612         \else
613             \PackageWarning{exesheet}{Invalid option `noteragged=twoside'
614              when the document \MessageBreak is not in two-side mode}
615         \fi
616     }{% else
617     \PackageWarning{exesheet}{The value `\exesheet@noteragged'
618          is not supported by the `noteragged' option}
619     }}}}}
620 }
621
```

The scale control option relies on calculations with *lengths*, which need to have a *global* scope. To achieve this, we first define the macros `\gsetlength` and `\gaddtolength`. These macros include % symbols at the end of lines to avoid expanded blank spaces.

For questions, assigned points will be added in `\sum@pts`, while for exercises, points accumulate in `\sum@exe`. These lengths are compared against `\exe@total` and `\sheet@total`. The `\exe@check` macro validates the calculations of the previous exercise when triggered by `\points`, `\totalexe` or `\totalpoints` macros. It is also invoked within `\exs@process@checkpts` at the document's end for a final check on the last exercise.

```
622 \newlength{\sheet@total}
623 \newlength{\sum@exe}
624 \newlength{\exe@total}
625 \newlength{\sum@pts}
626 \def\exe@currentlabel{none}
627 \newboolean{scale@valid}
628
629 \gdef\gsetlength#1#2{% for obtaining global length values
630     \begingroup
631        \setlength\skip@{#2}% local assignment to a scratch register
632        \global#1=\skip@%     global assignment to #1
633     \endgroup               % \skip@ is restored at the end of the group
634 }
635
636 \gdef\gaddtolength#1#2{% percent symbol necessary here!
637     \begingroup
638        \setlength\skip@{#1}%
639        \addtolength\skip@{#2}%
640        \global#1=\skip@%
641     \endgroup
642 }
643
644 \def\exe@check{
645     \ifthenelse{\lengthtest{\sum@pts = 0pt}}{
646     % do not check, no points or first exercise begins
647     \ifthenelse{\equal{\exe@currentlabel}{none}}{}{
648        \PackageWarningNoLine{exesheet}{\exe@currentlabel:
649           \the\exe@total}}
650     }{
651        \ifthenelse{\lengthtest{\exe@total = \sum@pts}}{
652           \PackageWarningNoLine{exesheet}{\exe@currentlabel:
653              Sum of points \the\exe@total\space is valid}
654        }{
655        \PackageWarningNoLine{exesheet}{\exe@currentlabel:
656           Sum of points is \the\sum@pts\space
657           instead of \the\exe@total}
658        \setboolean{scale@valid}{false}
659        }
660     }
661 }
662
663 \def\exs@process@checkpts{
664     \ifexesheet@checkpts
```

```
665          \ifthenelse{\lengthtest{\sheet@total = 0pt}}{
666              \PackageWarningNoLine{exesheet}{Option checkpts is true,
667                  \MessageBreak
668                  but \string\totalsheet\space is missing
669                  in the preamble. \MessageBreak
670                  See documentation}
671          }{}
672          \gsetlength{\sum@exe}{0pt}
673          \gsetlength{\exe@total}{0pt}
674          \gsetlength{\sum@pts}{0pt}
675          \setboolean{scale@valid}{true}
676          \AtEndDocument{% final checking (global)
677              \ifthenelse{\equal{\exe@currentlabel}{none}}{
678                  \ifthenelse{\lengthtest{\sum@pts = 0pt}}{
679                      \PackageWarningNoLine{exesheet}{checkpts: No points displayed}
680                  }{
681                      \ifthenelse{\lengthtest{\sheet@total = \sum@pts}}{
682                          \PackageWarningNoLine{exesheet}{Total:
683                              Sum of points \the\sheet@total\space is valid}
684                      }{
685                          \PackageWarningNoLine{exesheet}{Total:
686                              Sum of points is \the\sum@pts\space
687                              instead of \the\sheet@total}
688                  }}
689              }{% last exercise and final checking
690                  \exe@check
691                  \ifthenelse{\lengthtest{\sum@exe} = 0pt}{
692                      \PackageWarningNoLine{exesheet}{checkpts: No points displayed}
693                  }{
694                      \ifthenelse{\lengthtest{\sheet@total = \sum@exe}}{
695                          \PackageWarningNoLine{exesheet}{Total:
696                              Sum of points \the\sheet@total\space is valid}
697                      }{
698                          \PackageWarningNoLine{exesheet}{Total:
699                              Sum of points is \the\sum@exe\space
700                              instead of \the\sheet@total}
701                          \setboolean{scale@valid}{false}
702                      }
703                      \ifthenelse{\boolean{scale@valid}}{
704                          \PackageWarningNoLine{exesheet}{
705                              Marking scheme checked without errors}
706                      }{
707                          \PackageWarningNoLine{exesheet}{
708                              Marking scheme checked with ERRORS! See above}
709                      }
710                  }
711              }
712          }
713      \fi
714 }
715
```

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.

```
716 \newcommand*{\check@points}[1]{%
717     \ifexesheet@checkpts%
718         \exe@check% checks the previous exercise
719         \gdef\exe@currentlabel{\exe@label}% for the upcoming exercise
720         \gsetlength{\sum@pts}{0pt}%
721         \gsetlength{\exe@total}{#1pt}%
722         \gaddtolength{\sum@exe}{#1pt}%
723     \fi%
724 }
725
```

`\points`

```
726 \definecolor{pointscolor}{named}{red}
727 \newcommand{\pointsstyle}{%
728     \small\mdseries\sffamily\color{pointscolor}\fbox}
729 \newcommand*{\points}[1]{%
730     \ifthenelse{\boolean{exesheet@questions}}{\hfill
731         \pointsstyle{#1~%
732             \ifthenelse{\lengthtest{#1pt < 2pt}}{\pointname}{\pointsname}}
733         \check@points{#1}
734     }{}
735 }
736
```

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`

```
737 \definecolor{ptscolor}{named}{red}
738 \newcommand{\ptsstyle}[1]{%
739     \footnotesize\centering\sffamily\color{ptscolor} (#1)}
740 \newcommand*{\ptsmark}[1]{%
741     \ifthenelse{\lengthtest{#1pt < 2pt}}{#1 \ptname}{#1 \ptsname}}
742 \newcommand*{\pts}[1]{%
743     \ifexesheet@pts%
744         \mbox{}%
745         \marginpar{\hspace{0pt}\ptsstyle{\ptsmark{#1}}}%
746         \ifexesheet@checkpts%
747             \gaddtolength{\sum@pts}{#1pt}%
748         \fi%
749     \fi%
750     \ignorespaces
751 }
752
```

`\totalexe` 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.

```
753 \definecolor{markingcolor}{named}{red}
754 \newcommand{\markingstyle}[1]{\footnotesize\sffamily%
755     \centering\color{markingcolor}\textbf{#1}}
756     % inner arguments enable the implementation of boxed styles
757 \newlength{\ptsboxlength}
758 \setlength{\ptsboxlength}{3.1em}
759 \cornersize{1}
760 \newcommand*{\totalexe}[1]{%
761     \ifexesheet@pts%
762         \mbox{}%
763         \marginpar{\hspace{0pt}\markingstyle{\ovalbox{%
764             \makebox[\ptsboxlength]{\ptsmark{#1}}}}}%
765         \check@points{#1}%
766     \fi%
767     \ignorespaces
768 }
769
```

```
770 \newcommand*{\totalsheet}[1]{
771     \gsetlength{\sheet@total}{#1pt}
772 }
773
```

\note
\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. 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`).

```
774 \definecolor{notecolor}{rgb}{0.0, 0.4, 0.0} % kind of dark green
775 \newcommand{\notestyle}[1]{\footnotesize\sffamily\color{notecolor} #1}
776 \newcommand{\note@marginpar}[1]{%
777     \if@twoside%
778         \marginpar[\noteraggedleft #1]{\noteraggedright #1}%
779     \else%
780         \marginpar{\noteragged #1}%
781     \fi%
782 }
783 \newcommand{\@note}[2][]{%
784     \ifexesheet@pts%
785         \mbox{}%
786         \note@marginpar{%
787             \ifthenelse{\equal{#1}{}}{}{{%
788                 \noindent\hspace{0pt}\markingstyle{#1}\\}}%
789             \ifthenelse{\boolean{exesheet@notes}}{%
790                 \noindent\hspace{0pt}\notestyle #2%
791             }{}%
792         }%
793         \ifexesheet@checkpts%
794             \ifthenelse{\equal{#1}{}}{}{%
795                 \gaddtolength{\sum@pts}{#1pt}}%
```

```
796          \fi%
797      \fi%
798      \ignorespaces
799 }
800 \newcommand{\@@note}[1]{%
801      \ifexesheet@pts%
802          \mbox{}%
803          \marginpar{\noindent\hspace{0pt}\markingstyle{#1}}%
804          \ifexesheet@checkpts%
805              \gaddtolength{\sum@pts}{#1pt}%
806          \fi%
807      \fi%
808      \ignorespaces
809 }
810 \newcommand{\note}{\@ifstar{\@@note}{\@note}}
811
```

```
812 \newcommand{\totalpoints}{%
813      \ifthenelse{\boolean{exesheet@pts}}{\totalexe}{\points}}
814
```

7.8 The `correct` option and other (deprecated) commands

```
815 \def\exs@process@correct{
816      \ifthenelse{\equal{\exesheet@correct}{false}}{% do nothing
817      }{% else
818      \@ifpackageloaded{schooldocs}{
819          \ifthenelse{\equal{\exesheet@correct}{true}}{
820              \correct
821          }{% else
822          \ifthenelse{\equal{\exesheet@correct}{conditional}}{
823              \ifexesheet@answers \correct \fi
824          }{}}
825      }{
826          \PackageWarningNoLine{exesheet}{The 'correct' option requires
827              \MessageBreak
828              the 'schooldocs' package to be loaded}
829      }}
830 }
831
```

For the time being, the following macros are kept for compatibility reasons.

```
832 \newcommand{\questionsonly}{
833      \PackageWarning{exesheet}{The command \string\questionsonly\space
834          is deprecated; \MessageBreak
835          use the package option 'output=questions' instead}
836      \renewcommand\exesheet@output{questions}
837      \exs@process@output
838 }
839 \newcommand{\answersonly}{
840      \PackageWarning{exesheet}{The command \string\answersonly\space
```

```
841        is deprecated; \MessageBreak
842        use the package option 'output=answers' instead}
843    \renewcommand\exesheet@output{answers}
844    \exs@process@output
845 }
846 \newcommand{\displaypts}{%
847    \PackageWarning{exesheet}{The command \string\displaypts\space
848        is deprecated; \MessageBreak
849        use the package option 'display=pts' instead}
850    \renewcommand\exesheet@display{pts}
851    \exs@process@display
852 }
853 \newcommand{\displaypoints}{%
854    \PackageWarning{exesheet}{The command \string\displaypoints\space
855        is deprecated; \MessageBreak
856        use the package option 'display=pts' instead}
857    \renewcommand\exesheet@display{pts}
858    \exs@process@display
859 }
860 \newcommand*{\displaynotes}[1][\RaggedLeft]{%
861    % \renewcommand{\noteragged}{#1} no effect now!
862    \PackageWarning{exesheet}{The command \string\displaynotes\space
863        is deprecated; \MessageBreak
864        use the package option 'display=notes' instead}
865    \renewcommand\exesheet@display{notes}
866    \exs@process@display
867    \renewcommand{\noteragged}{#1}
868 }
869 \newcommand*{\displaynotesright}[1][\RaggedRight]{%
870    % \renewcommand{\noteragged}{#1} no effect now!
871    \PackageWarning{exesheet}{The command \string\displaynotes\space
872        is deprecated; \MessageBreak
873        use the package options 'display=notes, margin=right' instead}
874    \renewcommand\exesheet@display{notes}
875    \exs@process@display
876    \renewcommand\exesheet@margin{right}
877    \renewcommand{\noteragged}{#1}
878 }
879
880 \PackageInfo{exesheet}{The environment 'tablenum' is deprecated
881    \MessageBreak and has been replaced by 'tablenum1'.
882    \MessageBreak The options 'notoc' and 'nosetlist'
883    \MessageBreak are no longer supported\@gobble}
884    % \@gobble suppresses the line number here
885 ⟨/package⟩
```