The assign package[1]

Benjamin Bernard[2]

Abstract

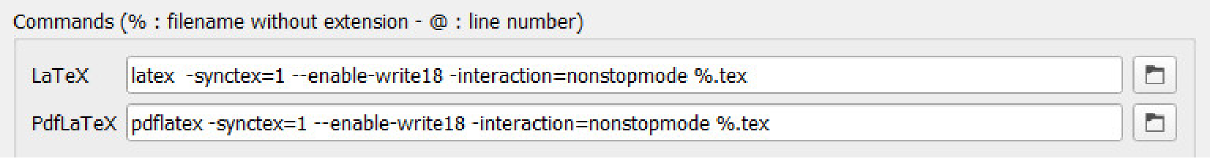
This is a LaTeX document class to write problem sets, assignments, exams, or questions banks. This document class can generate multiple pdfs for an assignment/exam, its solutions, and its grading scheme from a single source file. Points inserted into the grading scheme are added up automatically so that their sum can be displayed in the problem statements. To support the use as a question bank, problems can be filtered based on labels, difficulty, or quality. Many LaTeX editors provide a feature that allows the user to jump to the exact source code of a corresponding location in the pdf. These features are preserved—contrary to many packages written for a similar purpose.

1 Instructions for the impatient

The environments provided by this document class are `problem`, `subproblems`, and `subsubproblems`. Environment `problem` behaves similarly to a theorem environment. Environments `subproblems` and `subsubproblems` behave like an `enumerate` environment except that you use `\question` and `\answer` instead of `\item` to start a new item in the list. The easiest way to familiarize yourself with their usage is by looking at the examples in Sections 1.3 and 1.4. Whether the problem descriptions or their solutions are displayed is governed by the document options described in Section 1.2. The essential information is thus contained within Section 1, whereas Section 2 provides more details and customization options.

1.1 Installation: enabling write18

The multi-pass options require that write18 is enabled by adding `--enable-write18` on MiKTeX or `--shell-escape` on TeX Live and MacTeX to the *latex* or *pdflatex* call. In most LaTeX editors, this option will be available in the configurations or preferences menu. Write18 is enabled by default on Overleaf. Here is an example of the configuration menu on Texmaker, running on MiKTeX:



Commands (% : filename without extension - @ : line number)

LaTeX `latex -synctex=1 --enable-write18 -interaction=nonstopmode %.tex`

PdfLaTeX `pdflatex -synctex=1 --enable-write18 -interaction=nonstopmode %.tex`

1.2 Document class options

Write your document with one of the following document class options:

- `assign`. The problem descriptions are displayed.
- `points`. The problem descriptions and the points are displayed.
- `solution`. The solutions are displayed.
- `grading`. The solutions, the points, and the instructions to the grader are displayed.
- `draft`. Everything is displayed and the `draft` option of the `article` document class is enabled.

[1] This file has version number v0.89, last revised on June 25, 2024.
[2] Research assistance from 王芝雲 (Zoe Wang) is greatly appreciated.

Any of the above options require a single pass-through by LaTeX. Once you have finished writing the document, enable one of the multi-pass options to generate three pdfs from the source *filename.tex* at once: *filename.pdf* containing the assignment/exam, *filename_solutions.pdf* containing the solutions, and *filename_grading.pdf* containing the grading scheme. The two multi-pass options are:

- `exam`. Points are displayed on the assignment/exam.
- Without any of the above document options, points are not shown on the assignment/exam.

Overleaf uses a different internal filenaming system. To use the multi-pass options on Overleaf, you must additionally provide the document class option `overleaf`. This option works as of writing this document, but the compatibility may break if Overleaf's filenaming system changes.

Option `French` enables French indentation—although that is not recommended for a document full of enumerations. Option `qbank` enables a filtering system to manage a question bank; see Section 2.4 for details. Finally, any options supported by the `article` document class are supported as well.

Warning: It is imperative that you write your problem set in one of the single-pass modes. The multi-pass modes start processes *pdflatex filename_solutions* and *pdflatex filename_grading* in the background. However, because your LaTeX editor is aware of only the process *(pdf)latex filename*, any error messages thrown in the background processes are not directed towards your LaTeX editor. Such errors may cause the editor to freeze, run indefinitely, or it may cause other unexpected errors.

1.3 Minimal working example

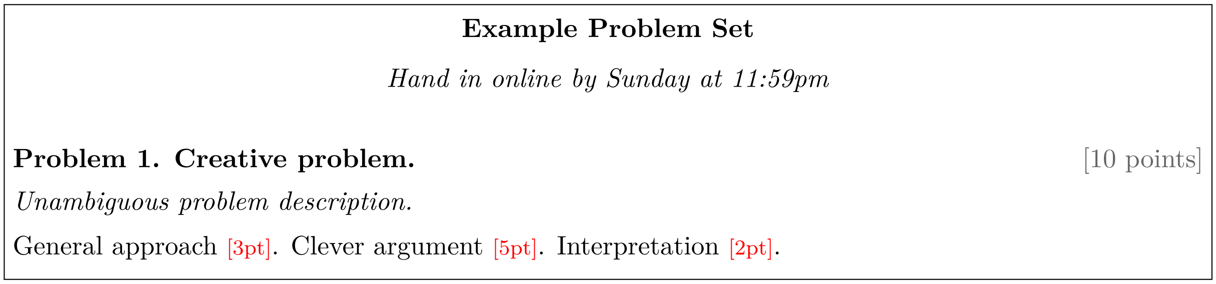
The following is a minimal working example with default outputs for the `draft` and `exam` document options. See Sections 1.4 and 2 below for ways to customize the output.

```latex
\documentclass[draft]{assign}
\title{Example Problem Set}
\instructions{Hand in online by Sunday at 11:59pm}

\begin{document}
\maketitle

\begin{problem}[title=Creative problem]
  Unambiguous problem description.
  \answer
  General approach\pt[3]. Clever argument\pt[5]. Interpretation\pt[2].
\end{problem}

\end{document}
```

Add a title with `\title` and `\maketitle` as usual. The macro `\instructions` allows you to add instructions to the students that are displayed only on the assignment/exam, but not on the solutions.

The main environment used to typeset assignments or exams is the `problem` environment. Everything between `\begin{problem}` and `\answer` is typeset on the assignment/exam, everything between `\answer` and `\end{problem}` is typeset on the solutions and the grading scheme. The environment takes a list of options passed as key-value pairs; see Section 2 for a full list of available options. Add points to the grading scheme with the command `\pt[<num>]`. Without the optional argument, `\pt` adds a single point. Note that the points are issued in the `\answer` part of the problem. The points are totaled up and displayed next to the problem's title (two compilations required in single-pass modes).

In `draft` mode, by default the problem description is typeset in italics to separate it from its solution.



Example Problem Set

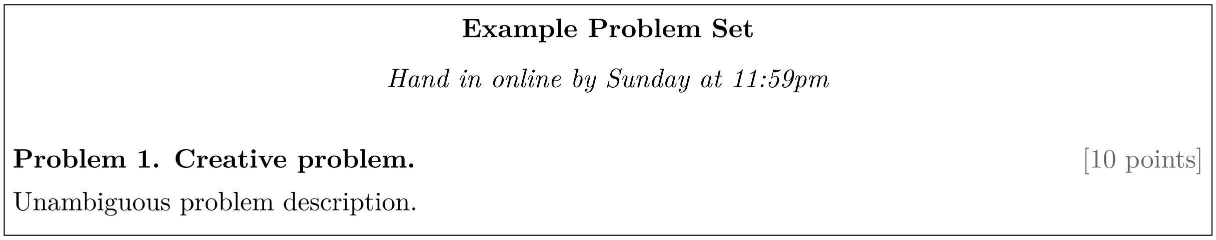
Hand in online by Sunday at 11:59pm

Problem 1. Creative problem. [10 points]

Unambiguous problem description.

General approach [3pt]. Clever argument [5pt]. Interpretation [2pt].

The multi-pass option `exam` generates three pdfs corresponding to the document options `points`, `solution`, and `grading`, shown below in order. In `points` mode, only the problem description is shown.



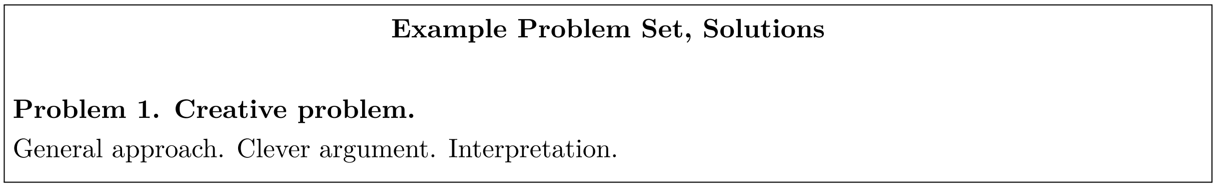
Example Problem Set

Hand in online by Sunday at 11:59pm

Problem 1. Creative problem. [10 points]

Unambiguous problem description.

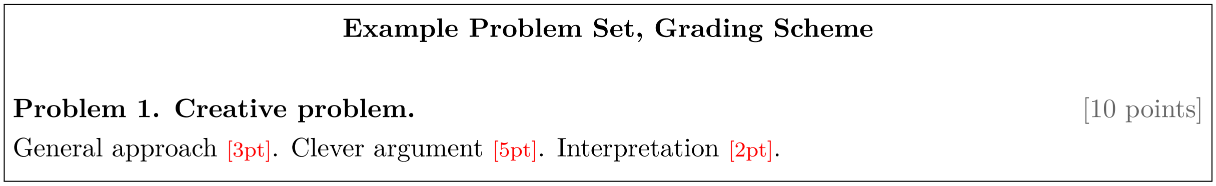
In `solution` and `grading` mode, instructions and problem descriptions are suppressed, and "Solutions" and "Grading Scheme" are added to the title, respectively. Points are suppressed in `solution` mode.



Example Problem Set, Solutions

Problem 1. Creative problem.

General approach. Clever argument. Interpretation.



Example Problem Set, Grading Scheme

Problem 1. Creative problem. [10 points]

General approach [3pt]. Clever argument [5pt]. Interpretation [2pt].

1.4 A more elaborate example

The following is a more elaborate example that involves subproblems and customization of the output.

```latex
\documentclass[draft]{assign}

\solutionsetup{show question, show points, color=blue, question font=\normalshape,
    header=\emph{Solution:\space}}
\pointsetup{margin, label=pt, sep={}, font=\small, hierarchy=lowest}

\begin{document}
\begin{problem}[title=Generic problem, source={generic textbook}]
  General description of the problem.

  \begin{subproblems}
```

```
    \question General question. \hint{some useful hint.}
    \answer Some setup\pt[2].
    \[ \text{Some mathematical derivation}\pt[3]. \]
  \end{subproblems}

  Restrict attention to a special case.

  \begin{subproblems}
    \question Another question with two subcases.\label{q:SpecialCase}

    \begin{subsubproblems}
      \question Simple case.
      \answer Answer\pt.
      \question More difficult case.
      \answer Answer\pt[2].
    \end{subsubproblems}

    \question Provide some interpretation to the answers in~\ref{q:SpecialCase}
    \answer We distinguish the two cases.
    \begin{enumerate}
      \item Simple insight\pt.
      \item More advanced insight\pt. Bonus for particularly clever insight\bonus[2].
    \end{enumerate}
  \end{subproblems}
\end{problem}

\end{document}
```

The generated output looks as follows.

Problem 1. Generic problem (generic textbook).

General description of the problem.

a. General question. [5pts]

Hint: a useful hint.

Solution: Some setup [2pt].

$$\text{Some mathematical derivation [3pt].}$$

Restrict attention to a special case.

b. Another question with two subcases.

(i) Simple case. [1pt]

Solution: Simple answer [1pt].

(ii) More difficult case. [2pts]

Solution: More complicated answer [2pt]. [Additional instructions to the TA/grader.]

c. Provide some interpretation to the answers in **b.** [2pts]

Solution: We distinguish the two cases.

(i) Simple insight [1pt].

(ii) More advanced insight [1pt]. Bonus for particularly clever insight [2pt].

First, the commands `\pointsetup` and `\solutionsetup` in the preamble customize the appearance of problems and their solutions in the entire document. The options `show question` and `show points` enable that the problem descriptions and the points, respectively, are included on the solutions and the grading scheme. The option `color=blue` typesets the solutions in blue, and the option `header=<code>` prefaces the solutions to each problem with `<code>`. The option `question font=\normalshape` typesets the question upright rather than italicized in the `solution` and `draft` modes.

The option `margin` passed to `\pointsetup` displays the points in the margin of the problem. By default, both the totals for each subproblem and the total for each problem are displayed. The option `hierarchy=lowest` suppresses this behavior and points are displayed only in the lowest hierarchy of problems. The options `label=pt`, `sep={}`, and `font=\small` adjust the size and the label of the points.

Different from the minimal working example, this problem is structured with environments `subproblem` and `subsubproblems`. Both environments work like an `enumerate` environment, except that you use `\question` and `\answer` instead of `\item` to start a new item. Everything between a `\question` and its following `\answer` command is displayed on the assignment/exam, whereas anything between an `\answer` and the next `\question` command (or the end of the environment) is shown on the solutions and the grading scheme. Note that an `\answer` command has to be issued only on the lowest hierarchy of problems: if a problem is divided into subproblems, then the `problem` itself does not require an `\answer`.

The counters of `subproblems` and `subsubproblems` reset only at the the beginning of the next `problem` and `subproblem`, respectively. Thus, a list of subproblems or subsubproblems can be interrupted with additional instructions simply by ending and restarting the environment. You can manually set the counter of `subproblems` or `subsubproblems` to `<int>` by passing `<int>` as an optional argument to either environment. To manually change the problem counter, use `\setcounter{problem}{<int>}`.

Both environments are compatible with the *enumitem* package. The environments `subproblems` and `subsubproblems` inherit spacing and labels from the first and second level of the `enumerate` environment, respectively. Thus, you can adjust spacing as you would with the *enumitem* package.

Bonus points can be added with `\bonus[<num>]`. Without the optional argument, `\bonus` will add a single bonus point. Bonus points are excluded from the point totals. Instructions to the TA/grader can be added with the `\ta` command. Those will be displayed only on the grading scheme. Hints provided with the `\hint` command are italicized and shown only if the question is displayed.

1.5 Figures and tables

Figures and tables can be added within a `problem` environment as one would in any other document class. If they appear before the `\answer` command, they are displayed in the assignment/exam. If they appear afterwards, they are displayed in the solutions and the grading scheme.

Warning: Outside of a `problem` environment, figures and tables require an extra mandatory argument `assign`, `solution`, or `both` that indicates where the figure should be displayed. For example:

```
\begin{figure}[h]{solution}
  \rule{5pt}{5pt}
  \caption{What a graph!}
\end{figure}
```

1.6 Compatibility issues

Compatibility issues may arise with any packages that write text into auxiliary files. As per the writing of this document, this document class is compatible with the *hyperref* package, as well as plotting graphs with *GNUplot*; see Section 22.6 in version 3.1.10 of the Ti*k*Z & PGF manual.

2 Full description of environments and commands

The description of document options is complete in Section 1.2 and is omitted here.

2.1 Problems and subproblems

The main syntax of the `problem` environment is the following:

```
\begin{problem}[<options>] <body> \end{problem}
```

The environment `<body>` can be divided into a problem description and its solution with the `\answer` command as in Section 1.3, or it can be structured with `subproblems` as in Section 1.4.

The `problem` environment is available also as a starred version `problem*` that suppresses the problem counter. If no title is given to the problem, `problem*` suppresses the entire problem header.

Both environments take one optional argument `[<options>]`, where `<options>` is a comma-separated list of key-value pairs that govern the appearance of the problem header and provide filter criteria for use in a question bank. They key-value pairs governing the header are the following:

`inspiration = <string>` appends `(inspired by <string>)` to the problem's title.

`source = <string>` appends `(<string>)` to the problem's title.

`title = <string>` adds a title `<string>` to the problem.

`version = <string>` appends `(<string>)` to the problem's title in `qbank` mode. This allows you to keep several versions of a problem in your question bank that are easily distinguished by their title. The distinction in the title is not present in all other document modes, i.e., if you compile the problem in `exam` mode, the distinction will be invisible to the students.

If both `version` and `source/inspiration` are provided, they are displayed as `(<source/inspiration>, <version>)`. The key-value pairs that support filtering in a question bank are the following:

`difficulty = {<concept>, <math>}` assigns two integers `<concept>` and `<math>` to the problem, allowing you to keep track of and filter based on a problem's difficulty.

`labels = {<list of strings>}` adds each element of the comma-separated `<list of strings>` as a label to the problem to enable filtering based on labels in `qbank` mode; see Section 2.4.

`quality = {<instructive>, <fun>}` assigns two integers `<instructive>` and `<fun>` to the problem, allowing you to keep track of and filter based on a problem's quality.

The main syntax of the `subproblems` and `subsubproblems` environment is the following:

```
\begin{subproblems}[<int>] <body> \end{problem}
```

Items are added to the list of sub/subproblems with `\question` and `\answer` commands. Each `\question` is directly followed by its `\answer` except if the `\question` is further divided into `subsubproblems`; see Section 1.4 for an example of both. The environment `<body>` must contain at least one `\question`, same as any `enumerate` environment must contain at least one `\item`.

The optional argument `<int>` sets the counter of the first sub/subproblem in the list to `<int>`.

Finally, the following commands are useful to write problem sets and grading schemes.

`\bonus [<float>]` indicates that `<float>` bonus points are to be awarded for a certain argument. Bonus points are not added to point totals for any sub/sub/problems.

`\hint {<text>}` displays `Hint: <text>` after the current sub/sub/problem's description.

`\instructions {<text>}` displays `<text>` on the assignment/exam; intended for instructions to the students.

`\maketitle` is a redefinition of its variant in the `article` class that will modify the title for solutions and grading schemes. Use immediately after `\begin{document}` as you would normally.

`\pt [<float>]` indicates that an argument is `<float>` points worth. All points issued through `\pt` within the same sub/sub/problem are added up and stored in the auxiliary file *filename.pt* so that their totals can be included in the problem description. In any single-pass document mode, two compilations are required to obtain the correct totals. In multi-pass modes, a single compilation is sufficient. See Section 2.2 for how to customize the displayed points.

`\ta {<text>}` displays `[<text>]` on the grading scheme, intended for instructions to the TA/grader.

`\title {<text>}` sets the title of the document to be typeset with `\maketitle`.

Spacing around the `\pt` and `\bonus` commands is optimized for use before punctuation, i.e., include `\pt` and `\bonus` at the end of a sentence or subclause.

The following three commands allow you to issue arbitrary blocks of code only in the assignment/exam, the solutions, or the draft. This may be helpful to improve page breaks or to customize title pages.

`\assign {<code>}` executes `<code>` only in the assignment/exam.

`\draft {<code>}` executes `<code>` only in `draft` mode.

`\solution {<code>}` executes `<code>` only on the solutions and the grading scheme.

2.2 Customizing appearance

Customization is provided through the four commands `\gradingsetup`, `\pointsetup`, `\questionsetup`, and `\solutionsetup`, each taking a comma-separated list of key-value pairs `<options>`.

`\gradingsetup {<options>}` governs the appearance of the grading scheme. Its available `<options>` are:

`color = <color>` sets the color of the instructions to the TA/grader to `<color>`.

`title = <string>` appends `<string>` to the document tile of the grading scheme, separated by a comma. The initial value of `<string>` is “Grading Scheme”.

`\pointsetup {<options>}` governs the appearance of the points. Its available `<options>` are:

`bonus color = <color>` sets the color of bonus points in the grading scheme to `<color>`.

`font = <font>` sets the font of point totals to `<font>`.

`hierarchy = <option>` determines where point totals are displayed. Admissible values for `<option>` are any positive integer, in which case point totals are shown for the top `<option>` hierarchy of problems, or `lowest`, in which case point totals are shown only for the lowest hierarchy.

`label = <string>` labels point totals with label `<string>`. The initial value is `point`.

`margin = <bool>` indicates whether point totals should be typeset in the margin. The default value (if the key is specified without value) is `true`, and the initial value is `false`.

`pt color = <color>` sets the color of points from the `\pt` command to `<color>`.

`sep = <string>` separates the label of point totals by `<string>`. The initial value is a space.

`totals color = <color>` sets the color of point totals to `<color>`.

`\questionsetup {<options>}` governs the appearance of problem descriptions. Its available `<options>` are:

`color = <color>` sets the color of problem descriptions to `<color>`.

`font = <font>` sets the font of problem descriptions to `<font>`.

`hint color` = `<color>` sets the color of hints to `<color>`.

`hint font` = `<font>` sets the font of hints to `<font>`.

`\solutionsetup {<options>}` governs the appearance of solutions. Its available `<options>` are:

`color` = `<color>` sets the color of solutions to `<color>`.

`font` = `<font>` sets the font of solutions to `<font>`.

`header` = `<code>` precedes any solution with `<code>`. This is used to begin each answer with *Solution:* in Section 1.4, but any other code could be added.

`indent answer` = `<bool>` indicates whether the first paragraph of each answer is indented if document option `French` is active. The default value and the initial value are both `true`.

`question color` = `<color>` sets the color of problem descriptions to `<color>` if `show question` is enabled.

`question font` = `<font>` sets the font of problem descriptions to `<font>` if `show question` is enabled.

`show points` = `<bool>` indicates whether point totals should be displayed on the solutions. The default value (if the key is specified without value) is `true`, and the initial value is `false`.

`show question` = `<bool>` indicates whether the questions are shown on the solutions and the grading scheme. The default value (if the key is specified without value) is `true`, and the initial value is `false`.

`title` = `<string>` appends `<string>` to the document tile of the solutions, separated by a comma. The initial value of `<string>` is “Grading Scheme”.

2.3 Spacing

White space before and after a problem header is determined by the lengths `\beforeheadersep` and `\afterheadersep`, respectively. White space between a question and its answer is the length `\answersep`. White space within `sub/subproblems` is inherited from the *enumitem* package and it is recommended that you change any spacing as instructed in the *enumitem* package description. Interline spacing is handled with the *setspace* package, margins are handled with the *geometry* package, and the spacing of tables and figures is handled with the *caption* package. The default values are the following.

```
% geometry
\RequirePackage[margin=1in, bottom=1.1in]{geometry}
% default lengths
\beforeheadersep \dimexpr\baselineskip minus 0.5\baselineskip\relax
\afterheadersep 0.8ex plus 0.2ex
\answersep 1.2ex plus 0.2ex
\parskip 0.8ex plus 0.1ex
% enumitem
\partopsep 0pt
\setlist{labelsep=0.5em, itemsep=0.3ex plus 0.1ex, parsep=\parskip}
\setlist[1]{leftmargin=1.76em, labelwidth=1.15em, topsep=0.4ex}
\setlist[2]{leftmargin=2.2em, topsep=0.5ex plus 0.1ex, listparindent=\parindent}
\setenumerate[1]{label=\textbf{\alph*.}, align=left}
\setenumerate[2]{label=\textbf{(\roman*)}, align=left}
% setspace
\linespread{1.09}
% caption
\captionsetup{font=small, label font=bf, margin={1.76em, 0pt}, belowskip=-0.6em}
```

Alternatively, you can manage white space with the command `\setspace{<options>}`, where `<options>` is a comma-separated list of key-value pairs from the following list:

`above caption skip = <length>` sets the space between a figure/table and its caption to `<length>`.

`afterheadersep = <length>` sets the space after a problem header to `<length>`.

`answersep = <length>` sets the space between a problem and its answer to `<length>`.

`beforeheadersep = <length>` sets the space before a problem header to `<length>`.

`below caption skip = <length>` sets the space between the caption of a figure/table and the main text to `<length>`.

`displayskip = <length>` sets the space before and after a displayed equation to `<length>`.

`itemsepi = <length>` sets the space between two consecutive subproblems or items in the first level of an enumerate and itemize environments to `<length>`.

`itemsepii = <length>` sets the space between two consecutive subsubproblems or items in the second level of an enumerate and itemize environments to `<length>`.

`labelsepi = <length>` sets the space between the label and the paragraph it precedes of a subproblem or an item in the first level of an enumerate and itemize environments to `<length>`.

`labelsepii = <length>` sets the space between the label and the paragraph it precedes of a subsubproblem or an item in the second level of an enumerate and itemize environments to `<length>`.

`labelwidthi = <length>` sets the width of the textbox containing the label of subproblems or items in the first level of an enumerate and itemize environments to `<length>`.

`labelwidthii = <length>` sets the width of the textbox containing the label of subsubproblems or items in the second level of an enumerate and itemize environments to `<length>`.

`leftmarginii = <length>` sets the left margin of subsubproblems and the second level of an enumerate and itemize environments to `<length>`.

`leftmargini = <length>` sets the left margin of floats like figures and tables, subproblems, and the first level of an enumerate and itemize environments to `<length>`.

`leftmarginii = <length>` sets the left margin of subsubproblems and the second level of an enumerate and itemize environments to `<length>`.

`linespread = <num>` sets the linespread of the *setspace* package to `<num>`.

`parindent = <length>` sets the indent of the first line of a paragraph to `<length>`.

`parsep = <length>` sets the space between two consecutive text paragraphs to `<length>`.

`parskip = <length>` sets the space between two consecutive text paragraphs to `<length>`.

`topsepi = <length>` sets the space between a list of subproblems or the first level of an enumerate and itemize environments and its surrounding paragraph to `<length>`.

`topsepii = <length>` sets the space between a list of subsubproblems or the second level of an enumerate and itemize environments and its surrounding paragraph to `<length>`.

The command `\setspace` handles some lengths slightly differently than *enumitem*'s `\setlist`:

- `leftmargini` governs both the left margin of enumerate and itemize environments as well as floats such as figures and tables. This ensures that figures are aligned over the enumerated list of answers.
- `parindent` sets both the indent of paragraphs within and outside of itemized lists.

- parskip and parsep are set equal to each other and they govern only the space between two consecutive text paragraphs. Changing either with the \setspace command adjusts answersep, afterheadersep, itemsepi/ii, and topsepi/ii so that the total white space governed by these keys remains unchanged.

2.4 Question banks

With document option qbank you can enable filtering of your question bank. Similarly to the draft mode, every command is typeset in qbank mode. Use the following three commands in the preamble to display only problems that match the filters.

\difficulty {concept<rel><int>, math<rel><int>} allows you to filter for a problem's difficulty along two dimensions, called concept and math. Here, <rel> is a binary relation among <, <=, =, >=, and > and <int> is an integer. It is possible to filter only for concept or math.

\filter [<rel>]{<list of labels>} filters problems based on the labels assigned to them, where <rel> is either and or or and <list of labels> is a comma-separated list of strings.

\quality {instructive<rel><int>, fun<rel><int>} allows you to filter for a problem's quality along two dimensions, called instructive and fun. Here, <rel> is a binary relation among <, <=, =, >=, and > and <int> is an integer. It is possible to filter only for instructive or fun.

Warning: The qbank mode breaks the feature of many LaTeX editors that locate the source code of any location in the pdf. To enable the filtering, the entire problem is placed within a group, which means that the source code locator will jump to the end of the problem instead. The idea is that the problems are written in draft mode and only when they are finished, they are stored in mode qbank.

2.5 Package dependencies

This document class is built on the article document class. Any features available in an article are available here. The following packages are loaded with options and an option clash will occur if you try loading those packages with a different set of options:

- [T1]{fontenc},
- [nodisplayskipstretch]{setspace},
- [protrusion=false]{microtype},

Moreover, packages geometry, lmodern, environ, enumitem, caption, xcolor, xspace, pgffor, pgfmath, and amsmath are loaded without options. You may load them again with options if you like.