The examz class*

Christopher McClain†

Released 2023/06/07

Abstract

The examz document class builds on the exam document class that was developed by Philip S. Hirschhorn. An author may use the class exactly as the exam class, but there are also additional features. The document class facilitates the writing of questions with random elements, the creation of multiple versions of an exam, and the use of separate files as question banks.

Contents

*This file describes version v1.0.0, last revised 2023/06/07.

†E-mail: christopher.mcclain@mail.wvu.edu

1 Introduction

1.1 About

Since 1994, Philip Hirschhorn's exam document class has provided authors and educators with a framework for writing exams that is both highly customizable and easy to use. The examz document class loads the exam class and adds some more features. Most of these new features require the counterz package.

First of all, the examz class may be loaded with new options that facilitate the use of commands from the counterz package for writing questions with random elements. Second, a new versions environment allows for the creation of multiple versions of an exam in the same document, alternating between exams and solutions, if desired. Third, the command `\question` may now be replaced by the command `\questionfrombank` which allows for random selection of questions from a user-defined question bank. Finally, a few simple macros make it even easier to modify the customized content in headers, footers, and special pages when creating new exams from previous documents.

Some commands in exam have been patched in order to distinctly label the questions in different exam versions and their solutions (so that LaTeX issues no warnings about multiply-defined labels). The optional environment coverpages has also been patched for two-sided documents so that the subsequent odd arabic numerals always appear on the front pages.

1.2 License

1.3 Installation

Run LaTeX on examz.ins to generate the file examz.cls, and copy it to your local texmf directory. Run (pdf)LaTeX on examz.dtx to generate the documentation examz.pdf. Typesetting the documentation requires the package hypdoc which is included in TeX distributions and at The Comprehensive TeX Archive Network.

2 User Guide

To use this document class, begin your document with the following line:

```
\documentclass{examz}
```

This document class loads the exam class and admits the options for that class. See the documentation for the exam class for details. The examz class also admits the additional options randomize, norandomize, prompt, and complete, all of which are discussed in this guide. This document class requires the counterz package, which is available at The Comprehensive TeX Archive Network.

In addition to its parent class exam and the counterz package, the examz class requires the following packages: xparse, environ, etoolbox, and makecmds (the latter two via counterz), all of which are included in most standard TEX distributions and at The Comprehensive TeX Archive Network.

2.1 Randomized Exams

The examz class loads the counterz package which provides commands for generating, saving, recalling, manipulating, and displaying random values for counters. These commands may be incorporated into questions to generate random exams. The process of saving and recalling counters may be explicitly coded in the document, as described in the counterz package documentation, but the examz class may instead be loaded with options to automate this process.

Loading the class with the option randomize will execute the commands `\randomizectr` and `\opencountersfile` from the counterz package. Loading the class with the option norandomize will execute the commands `\norandomizectr` and `\inputcountersfile`. If neither of these options are loaded, the default behavior is that of the option randomize. The reason for this is that if the document contains no randomly generated counters then this default choice will have no effect anyway, and if the document does contain randomly generated counters then the first typesetting must generate these counter values.

A third option prompt frees the user from the burden of manually changing between randomize and nonrandomize and also protects against accidental overwriting of previously generated counter values. This option will execute the command `\promptrandomexam`, which is an instance of the command `\promptrandomizectr` from the counterz package, followed by one of the commands `\opencountersfile` or `\inputcountersfile` as appropriate. The terminal dialogue of `\promptrandomexam` can be changed with the use of `\renewcommand`, if desired. (See Section 3 for the implementation of `\promptrandomexam`.) There are additional effects from this option with regard to the versions environment, described below.

`versions`

The versions environment uses a loop and a counter *version* to generate versions of the exam, incrementing the counter and updating the page numbers along the way. The version number may be displayed in headers or elsewhere with either `\theversion` or `\arabic{version}`. The environment takes an optional argument: the number of desired versions, with 1 being the default value. The environment is used as follows:

```
\begin{versions}[<optional number>]
   <exam content here>
\end{versions}
```

where *exam content* refers to the questions environment (as defined in the exam package), cover pages, and anything else after `\begin{document}` that is intended to appear in the exam. When the class is randomized with the option prompt, the number provided by the optional argument is not used, and so there is no reason to include it. Instead, a command `\promptversions` inputs the number of versions

using a terminal dialogue. (The dialogue can be changed by using `\renewcommand`, if desired. See Section 3.1 for the implementation of `\promptversions`.) When the exam is not randomized, the number of versions is inputted from the previously generated counters file, and so the prompt does not appear.

Recall that the examz class inherits the options answers and noanswers from the exam class. The examz class also has an additional option called complete. When loaded with this option, the document alternately prints without and with solutions. (Note: This feature only takes effect when using the versions environment.) Moreover, when the class is loaded with the option twoside (inherited from its grandparent class article), an extra blank page is inserted at the end of each exam when necessary to ensure that the next exam starts on a front page. An exam that begins with the line

```
\documentclass{11pt,twoside,addpoints,prompt,complete}
```

and uses the versions environment, for example, will ask the user whether to randomize, and if so, how many versions to print. Then the typeset document will include the requested number of versions, alternating between exams and solutions, with correct grade tables and no warnings about multiply-defined labels, and an extra blank page whenever an exam has an odd number of pages.

2.2 Question Banks

Within the questions environment, the examz package offers an alternative to the command `\question` that allows for the random selection of a question from a question bank. The command `\questionfrombank[`⟨*points*⟩`]{`⟨*filename*⟩`}` has two arguments, the first of which is optional. The first (optional) argument is the point value of the question, just as with the command `\question` from the exam class. The second argument is the name of a TEX file (without the .tex extension). The second argument may include a path. Suppose, for example, that the directory which contains the main exam document Test1.tex also contains a subdirectory named Problems containing the files Solve_Linear_Equation.tex and State_Capitals.tex. Then the code

`\questionfrombank`

```
\documentclass{examz}
\begin{document}
\begin{questions}
   \questionfrombank{Problems/Solve_Linear_Equation}
   \questionfrombank{Problems/State_Capitals}
\end{questions}
\end{document}
```

generates an exam with two questions, one randomly selected from each of the two specified files.

When several files are contained in the same directory, the repetition of the path name can be reduced by the command `\setquestionpath{`⟨*path name*⟩`}`, as illustrated by the following code:

`\setquestionpath`

```
\documentclass{examz}
\begin{document}
   \setquestionpath{Problems/}
\begin{questions}
   \questionfrombank{Solve_Linear_Equation}
   \questionfrombank{State_Capitals}
\end{questions}
\end{document}
```

The command `\setquestionpath` can be used both inside and outside of the questions environment, and can be used to change the path multiple times, if desired.

In order for a file to function as a question bank, it must conform to a specific (but very simple) structure. First of all, every question bank file begins with a
`\setnumberofquestions` command `\setnumberofquestions{`⟨*number*⟩`}`. Second, all of the questions are contained within a questionbank environment. Third, each individual question is contained within a qbitem environment. This structure is exhibited by the following code, which could be the entire contents of a file State_Capitals.tex inputted by the command `\questionfrombank`:

```
\setnumberofquestions{2}
\begin{questionbank}
\begin{qbitem}{1}
   What is the capital of Maine?
\end{qbitem}
\begin{qbitem}{2}
   What is the capital of Nebraska?
\end{qbitem}
\end{questionbank}
```

`questionbank` The questionbank environment randomly generates an integer between 1 and the number specified in the preceding command `\setnumberofquestions`. Each
`qbitem` instance of the environment qbitem`{`⟨*item number*⟩`}` compares its numerical argument ⟨*item number*⟩ to this randomly generated number and executes the environment's body of content only if the numbers match. Then the questionbank environment ends input so that nothing in the file after that environment will be included. Note that neither environment includes within its body the `\question` command; that command is already part of the command `\questionfrombank` which inputs the file. Also note that `\questionfrombank` sets the counter prefix to include the file name and exam version number, so authors should use the prefix versions of counters commands (e.g. `\xprovidecounter`, `\xarabic`, etc.) within qbitem environments. (See the counterz documentation for details.)

The qbitem environment allows for the inclusion of the various environments for parts and solutions that are provided by the exam class. Recall that each of the six solutions environments admits an argument to specify the amount of solution space. (The argument is optional for all except solutionbox. See the exam documentation for details.) The examz class offers a command
`\setsolutionspace` `\setsolutionspace{`⟨*length*⟩`}` and six analogous solutions environments that pass

qbsolution

qbsolutionbox
qbsolutionorbox
qbsolutionorlines
qbsolutionordottedlines
qbsolutionorgrid

⟨*length*⟩ to their exam counterparts. For example, the environment qbsolution is equivalent to the environment solution[⟨*length*⟩]. The advantage of this is that the command \setsolutionspace{⟨*length*⟩} can be included once in the file, above the questionbank environment, for example, and any subsequent adjustment of the length can be done for all of the question bank items simultaneously with a single edit rather than a comprehensive search and replace. Five other environments qbsolutionbox, qbsolutionorbox, qbsolutionorlines, qbsolutionordottedlines, and finally qbsolutionorgrid are similar analogues of the other exam class solutions environments. Just as their exam counterparts, these environments can also be used within the parts environment. The command \setsolutionspace may be used multiple times in the same file, including within the questionbank environment, if necessary.

2.3 Customization Macros

The exam document class provides the means for customizing headers, footers, special pages, etc. As an exam author makes changes to a document between exams, courses, or academic terms, the author may require frequent adjustment of certain standard content, such as the name of the exam or course in a custom header. The examz document class offers several macros to simplify such changes. For example, instead of explicitly including the text "MATH 101" in a header, an author can use the macro \coursename which can be set (and easily modified) at the beginning of the document. This is especially useful when designing for many similar courses a custom .cls file that loads the examz class and keeps the layout and customizations "behind the scenes". We first list and describe these macros and then provide an example of usage.

\instructorname
\setinstructorname

The default replacement text for \instructorname is "Instructor Name". Use the command \setinstructorname{⟨*text*⟩} to change this text.

\coursename
\setcoursename

The default replacement text for \coursename is "Course Name". Use the command \setcoursename{⟨*text*⟩} to change this text.

\examname
\setexamname

The default replacement text for \examname is "Exam Name". Use the command \setexamname{⟨*text*⟩} to change this text.

\termname
\settermname

The default replacement text for \termname is "Term Name". Use the command \settermname{⟨*text*⟩} to change this text.

\namespace
\setnamespace

The default expansion of \namespace is "Name: _____" which is given by the code Name:~\makebox[5cm]{\hrulefill}. Use the command \setnamespace{⟨*format*⟩} to change this format.

\instructions
\setinstructions

The default replacement text for \instructions is "Instructions Here". Use the command \setinstructions{⟨*text*⟩} to change this text, including one or more paragraphs as needed.

\covernoanswers

The command \covernoanswers can be used within the coverpages environment provided by the exam class or simply as the first page(s) of the exam. Its default

\setcovernoanswers expansion is empty. Use the command \setcovernoanswers{⟨*format*⟩} to change this format to include special headers and footers, instructions, grade/point tables, an instance of \newpage, etc.

\coveranswers The command \coveranswers can be used within the coverpages environment provided by the exam class or simply as the first page(s) of the exam. Its default ex-
\setcoveranswers pansion is \covernoanswers. Use the command \setcoveranswers{⟨*format*⟩} to change this format to include special headers and footers, instructions, grade/point tables, an instance of \newpage, etc.

\printcover The command \printcover executes either \coveranswers or \covernoanswers, depending on the value of the boolean printanswers. It can be used within the coverpages environment that is provided by the exam class or simply as the first page(s) of the exam.

\workspace The command \workspace is intended to provide additional work space on the exam, perhaps at the end of the questions. Its default expansion is empty. Use the
\setworkspace command \setworkspace{⟨*format*⟩} to add a bit of text, \newpage, or a custom header/footer. To include the content of \workspace only when the boolean
\printworkspace printanswers is FALSE, use the command \printworkspace.

One method of streamlining the work of developing many exams that use the same layout and format is to write a simple .cls file to use as a custom instantiation of the examz document class, perhaps even including frequently used packages:

```
\NeedsTeXFormat{LaTeX2e}
\ProvidesClass{MYexamz}
\DeclareOption*{\PassOptionsToClass{\CurrentOption}{examz}}
\ProcessOptions\relax
\LoadClass{examz}
\RequirePackage{mathtools,amssymb}
\pagestyle{headandfoot}
\header{\coursename}{\examname}{termname}
\setcovernoanswers{%
   \namespace

   \bigskip
   \instructions

   \bigskip
   \gradetable

   \firstpagefooter{Version \arabic{version}}{}{}
   \newpage
}%
\setworkspace{%
   \newpage
   \begin{center} Extra Work Space \end{center}
   \newpage
}%
\endinput
```

Then use the custom document class MYexamz for the exam:

```
\documentclass[addpoints,prompt,complete]{MYexamz}

\setcoursename{MATH 101}
\setexamname{Test 1}
\settermname{Spring 2023}
\setinstructions{%
   Read each problem carefully. Show all work.
}%

\begin{document}
\begin{versions}
   \printcover
   \setquestionpath{Problems/}
\begin{questions}
   \questionfrombank{Solve_Linear_Equation}
   \questionfrombank{State_Capitals}
\end{questions}
   \printworkspace
\end{versions}
\end{document}
```

To write a new exam Test 2 in the same course, simply change the argument of `\setexamname` from `Test 1` to `Test 2` and use the appropriate question banks. With random counters and well-developed question banks, writing exams for the next term may be as simple as changing the term name.

3 Implementation

The examz document class requires the packages environ, xpatch, and counterz, the last of which also loads the packages etoolbox and makecmds.

```
\RequirePackage{environ}
\RequirePackage{xpatch}
\RequirePackage{counterz}
```

This class inherits all of the options available to the exam class, including answers, noanswers, cancelspace, nocancelspace, and addpoints. The class also admits several additional options, the first of which is called complete. This option takes effect only when using the versions environment (see Section 3.1) and will otherwise be ignored. With this option selected, the document alternately prints each exam version with and without its solutions. The default value of the associated boolean variable is FALSE. Loading the option complete will change this value to TRUE. This option is intended to take the place of the options answers and noanswers and will override these two options if engaged.

```
\newbool{@examz@complete}
\boolfalse{@examz@complete}
\DeclareOption{complete}{\booltrue{@examz@complete}}
```

Additional options are based on features of the counterz package. The option randomize will cause the command \randprovidecounter (and its derivative commands) to generate new random values whereas the option norandomize will cause these commands to input their values from a file. Because this file may not exist before a first typesetting, the default option is randomize.

```
\randomizectr
\DeclareOption{randomize}{\randomizectr}
\DeclareOption{norandomize}{\norandomizectr}
```

Finally, the class may be loaded with an option called prompt. Because the manual change of the option randomize may be forgotten, resulting in the loss of counter values, the option prompt offers a terminal-based dialogue through which randomization can be decided. This action, which is executed at the beginning of the document (see below) is based on the command \promptranomizectr from the counterz package.

```
\newbool{@examz@prompt}
\boolfalse{@examz@prompt}
\DeclareOption{prompt}{\booltrue{@examz@prompt}}
\newcommand{\promptrandomexam}{%
   \promptrandomizectr[\EnterResponse]{%
      ^^J Enter 1 to randomize document.
      ^^J Enter 2 to update without new randomization.
   }{%
      1%
   }%
}%
```

We now load the exam document class with all of these options.

```
\DeclareOption*{%
   \PassOptionsToClass{\CurrentOption}{exam}
}%
\ProcessOptions\relax
\LoadClass{exam}
```

Next, we patch the coverpages environment. If the document is loaded with the option twoside (passed from the article class) and the coverpages environment produces an odd number of pages, then an extra blank page is inserted so that the odd arabic page numerals appear on the right page (i.e. front of the page).

```
\xpatchcmd{\endcoverpages}{\setcounter{num@coverpages}{\value{page}}}{%
   \ifbool{@twoside}{%
      \ifnumodd{\value{page}}{%
         % Do Nothing
      }{%
         \newpage
         \null
         \newpage
      }%
   }{%
```

```
      % Do nothing
   }%
   \setcounter{num@coverpages}{\value{page}}%
}{}{}%
```

If the class is loaded with the option prompt then we execute the command `\promptrandomexam`. If the user elects to not randomize the document, either by terminal input via `\promptrandomexam` or by instead loading the norandomize option, then we execute the command `\inputcountersfile`. (Note that this will produce an error if no counters file exists, so this option should not be used for a first typesetting.) If the user does elect to randomize the document, by terminal input or by loading the class with the option randomize or by loading the class with none of the options randomize, norandomize, or prompt, then we execute the command `\opencountersfile`.

```
\AtBeginDocument{%
   \ifbool{@examz@prompt}{%
      \promptrandomexam
   }{%
   }%
   \ifrandomizectr{%
      \opencountersfile
   }{%
      \inputcountersfile
   }%
}%
```

3.1 Randomized Exams

The counter *numversions* determines how many versions of the exam will be generated. The default value of *numversions* is 1.

```
\newcounter{numversions}
\setcounter{numversions}{1}
```

`\promptversions` The following command is used by the versions environment (described below) whenever the class is loaded with the prompt option. Note that the counter is saved to the counters file. (See the documentation for the counterz package.)

```
\newcommand{\promptversions}{%
   \typein[\NumberOfVersions]{%
      ^^J How many versions?
   }%
   \setcounter{numversions}{\NumberOfVersions}
   \savecounter{numversions}
}%
```

The counter *version* keeps track of the exam version when printing. The value of *version* is initialized as 1.

```
\newcounter{version}
\setcounter{version}{1}
```

We now patch the parent class exam so that corresponding questions in different versions will not be assigned the same question label. To accomplish this, we append to the question labels the value of *version*. We also append an S if the boolean *answers* (from the exam class) is TRUE, to distinguish between versions with and without solutions when the class is loaded with the option complete.

```
\xpatchcmd{\find@latestques}
   {@\arabic{question}}
   {@\arabic{version}\ifbool{printanswers}{S}{}@\arabic{question}}
   {}{}
\xpatchcmd{\decr@latest@ques}
   {question@}
   {question@\arabic{version}\ifbool{printanswers}{S}{}@}
   {}{}
\xpatchcmd{\chk@incompi}
   {question@}
   {question@\arabic{version}\ifbool{printanswers}{S}{}@}
   {}{}
\xpatchcmd{\questions}
   {@\arabic{question}}
   {@\arabic{version}\ifbool{printanswers}{S}{}@\arabic{question}}
   {}{}
\xpatchcmd{\parts}
   {@\arabic{question}}
   {@\arabic{version}\ifbool{printanswers}{S}{}@\arabic{question}}
   {}{}
\xpatchcmd{\subparts}
   {@\arabic{question}}
   {@\arabic{version}\ifbool{printanswers}{S}{}@\arabic{question}}
   {}{}
\xpatchcmd{\subsubparts}
   {@\arabic{question}}
   {@\arabic{version}\ifbool{printanswers}{S}{}@\arabic{question}}
   {}{}
\xpatchcmd{\setup@point@toks}
   {@\arabic{question}}
   {@\arabic{version}\ifbool{printanswers}{S}{}@\arabic{question}}
   {}{}
\xpatchcmd{\refto@index}
   {question@}
   {question@\arabic{version}\ifbool{printanswers}{S}{}@}
   {}{}
```

`\@examz@versions` The versions environment is defined via command `\@examz@versions`. The single argument of this command represents the user-provided exam content, including the questions environment and any pages before or after. If the class is loaded with the option prompt and the exam is randomized then the command first executes `\promptversions` to attain and save the value of the counter *numversions*. The main action of the command is to use a while loop to generate versions of the exam, updating the version and page numbers along the way. When the class

is loaded with the option complete, the document alternately prints without and with solutions, updating the version only after both printings. When the class is loaded with the option twoside, an extra blank page is inserted at the end of each version when necessary to ensure that the next version starts on a front page.

```
\newcommand{\@examz@versions}[1]{%
   \ifrandomizectr{%
      \ifbool{@examz@prompt}{%
         \promptversions
      }{%
         % Do Nothing
      }%
   }{%
      % Do Nothing
   }%
   \ifbool{@examz@complete}{%
      \noprintanswers
   }{%
      % Do Nothing
   }%
   \whileboolexpr{%
      test{\ifnumless{\value{version}}{1+\value{numversions}}}
   }{%
      \ifbool{@twoside}{%
         \ifnumodd{\value{page}}{%
            % Do Nothing
         }{%
            \newpage
            \null
            \thispagestyle{empty}
            \newpage
         }%
      }{%
         % Do Nothing
      }%
      \setcounter{page}{1}
      \setcounter{numquestions}{0}
      \setcounter{numparts}{0}
      \setcounter{numsubparts}{0}
      \setcounter{numsubsubparts}{0}
      \setcounter{numpoints}{0}
      \setcounter{numbonuspoints}{0}
      #1
      \newpage
      \notbool{@examz@complete}{%
         \addtocounter{version}{1}
      }{%
         \notbool{printanswers}{%
            \printanswers
         }{%
```

```
            \noprintanswers
            \addtocounter{version}{1}
         }%
      }%
   }%
}%
```

`versions` The versions environment is now defined using the previous command and the command `\Collect@Body`. The environment has an optional argument with which the user may specify the number of versions. If the exam is not randomized then the argument is ignored under the assumption that the needed value has been inputted from the counters file after a previous randomization. (If there is never any randomization, then there is no need for different versions, and the environment will execute with the default number of versions being one.) If the class is loaded with the option prompt, then this optional argument will be overridden by the number provided by the user via terminal prompt.

```
\newenvironment{versions}[1][1]{%
   \ifrandomizectr{%
      \setcounter{numversions}{#1}
      \savecounter{numversions}
   }{%
      % Do Nothing
   }%
   \Collect@Body\@examz@versions
}{%
   % Empty
}%
```

The exam document class exam prints in the console and log file some data that includes the number of questions, total points, etc. Here we add to that data the number of versions of the exam and whether the solutions were printed.

```
\AtEndDocument{%
   \typeout{%
      This document contains \thenumversions\space
      version\ifnumequal{\value{numversions}}{1}{}{s} of the exam
      \ifbool{@examz@complete}{%
         with and without solutions.
      }{%
         \ifbool{printanswers}{%
            with solutions.
         }{%
            without solutions.
         }%
      }%
   }%
}%
```

3.2 Question Banks

`\@examz@questionpath` `\setquestionpath` The default value of `\@examz@questionpath` is empty, but authors may use the command `\setquestionpath{`⟨*pathname*⟩`}` to change this.

```
\newcommand{\@examz@questionpath}{}
\newcommand{\setquestionpath}[1]{%
    \renewcommand{\@examz@questionpath}{#1}
}%
```

`\questionfrombank` The command `\questionfrombank[`⟨*points*⟩`]{`⟨*filename*⟩`}` uses the command `\question` from the exam class. The first (optional) argument is the point value of the question, and the second argument is the name of the question bank file (without the .tex extension). The input uses `\examz@questionpath`. In order to avoid a conflict between question banks that use the same counter names, `\counterprefix` is temporarily set to include both the name of the file and the number of the version. (See the counterz package for more information about `\setcounterprefix`.)

```
\newcommand{\questionfrombank}[2][]{%
    \question[#1]
    \setcounterprefix{#2_Version_\arabic{version}_}
    \input{\@examz@questionpath #2}
    \clearcounterprefix
}%
```

`\setnumberofquestions` The counter *@examz@qbsize* is an internal counter to represent the number of questions in a question bank. The command `\setnumberofquestions{`⟨*number*⟩`}` is used to set this value.

```
\newcounter{@examz@qbsize}
\newcommand{\setnumberofquestions}[1]{%
    \setcounter{@examz@qbsize}{#1}
}%
```

`questionbank` The questionbank environment begins by creating a random counter that is used to determine which question will be selected and ends with the command `\endinput` so that nothing after the environment appears in the question.

```
\newenvironment{questionbank}{%
    \xrandprovidecounter{Random_Question}{1}{\value{@examz@qbsize}}
}{%
    \endinput
}
```

`qbitem` The environment qbitem is a wrapper for each option in a questionbank environment. The argument is an integer value that is compared to the random counter value generated by questionbank. The body of the environment is the content of the question. Additional counters that are created and manipulated within qbitem should be handled by the commands provided in the counterz package in order to use the counter prefix that is created by the command `\questionfrombank`.

```
\NewEnviron{qbitem}[1]{%
\ifnumequal{\xvalue{Random_Question}}{#1}{%
\BODY
}{%
}
}
```

`\setsolutionspace` The command `\@examz@solutionspace` represents the amount of space allotted to solutions, per the various solutions environments that are defined by the exam document class. `\setsolutionspace{`⟨*length*⟩`}` is used to set this value.

```
\newcommand{\@examz@solutionspace}{1cm}
\newcommand{\setsolutionspace}[1]{%
   \renewcommand{\@examz@solutionspace}{#1}
}%
```

For each of the six solutions environments provided by the exam class there is an analogous solutions environment that automatically loads the value of `\SolutionSpace` for the argument.

`qbsolution` qbsolution is equivalent to solution[`\@examz@solutionspace`].

```
\NewEnviron{qbsolution}{%
   \begin{solution}[\@examz@solutionspace]
      \BODY
   \end{solution}
}%
```

`qbsolutionbox` qbsolutionbox is equivalent to solutionbox{`\@examz@solutionspace`}.

```
\NewEnviron{qbsolutionbox}{%
\begin{solutionbox}{\@examz@solutionspace}
\BODY
\end{solutionbox}
}%
```

`qbsolutionorbox` qbsolutionorbox is equivalent to solutionorbox[`\@examz@solutionspace`].

```
\NewEnviron{qbsolutionorbox}{%
\begin{solutionorbox}[\@examz@solutionspace]
\BODY
\end{solutionorbox}
}%
```

`qbsolutionorlines` qbsolutionorlines is equivalent to solutionorlines[`\@examz@solutionspace`].

```
\NewEnviron{qbsolutionorlines}{%
\begin{solutionorlines}[\@examz@solutionspace]
\BODY
\end{solutionorlines}
}%
```

qbsolutionordottedlines

qbsolutionordottedlines is equal to solutionordottedlines[\@examz@solutionspace].

```
\NewEnviron{qbsolutionordottedlines}{%
\begin{solutionordottedlines}[\@examz@solutionspace]
\BODY
\end{solutionordottedlines}
}%
```

qbsolutionorgrid

qbsolutionorgrid is equivalent to solutionorgrid[\@examz@solutionspace].

```
\NewEnviron{qbsolutionorgrid}{%
\begin{solutionorgrid}[\@examz@solutionspace]
\BODY
\end{solutionorgrid}
}%
```

3.3 Customization Macros

\instructorname
\coursename
\examname
\termname
\namespace
\instructions
\covernoanswers
\coveranswers
\workspace

The following commands define macros, with default values, for use in formatting headers, footers, and special pages. \covernoanswers and \workspace are empty by default, and \coveranswers is equal to \covernoanswers by default.

```
\newcommand{\instructorname}{Instructor Name}
\newcommand{\coursename}{Course Name}
\newcommand{\examname}{Exam Name}
\newcommand{\termname}{Term Name}
\newcommand{\namespace}{Name:~\makebox[5cm]{\hrulefill}}
\newcommand{\instructions}{Instructions Here}
\newcommand{\covernoanswers}{}
\newcommand{\coveranswers}{\covernoanswers}
\newcommand{\workspace}{}
```

\setinstructorname
\setcoursename
\setexamname
\settermname
\setnamespace
\setinstructions
\setcoveranswers
\setcovernoanswers
\setworkspace

The following commands allow a user to redefine the above macros.

```
\newcommand{\setinstructorname}[1]{\renewcommand{\instructorname}{#1}}
\newcommand{\setcoursename}[1]{\renewcommand{\coursename}{#1}}
\newcommand{\setexamname}[1]{\renewcommand{\examname}{#1}}
\newcommand{\settermname}[1]{\renewcommand{\termname}{#1}}
\newcommand{\setnamespace}[1]{\renewcommand{\namespace}{#1}}
\newcommand{\setinstructions}[1]{\renewcommand{\instructions}{#1}}
\newcommand{\setcovernoanswers}[1]{\renewcommand{\covernoanswers}{#1}}
\newcommand{\setcoveranswers}[1]{\renewcommand{\coveranswers}{#1}}
\newcommand{\setworkspace}[1]{\renewcommand{\workspace}{#1}}
```

\printcover

The command \printcover executes either \coveranswers or \covernoanswers, depending on the value of the boolean printanswers.

```
\newcommand{\printcover}{%
    \ifbool{printanswers}{\coveranswers}{\covernoanswers}
}%
```

\printworkspace

The command \printcover executes \workspace when the boolean printanswers is false and otherwise does nothing.

```
255 \newcommand{\printworkspace}{%
256     \ifbool{printanswers}{}{\workspace}
257 }%
```

4 Change History

v1.0.0

General: First public release 1

5 Index

Numbers written in italic refer to the page where the corresponding entry is described; numbers underlined refer to the code line of the definition; numbers in roman refer to the code lines where the entry is used.