A Hacker's Guide to the UC Exam Class

Michael JasonSmith

7th December 2004

Abstract

This guide will show you how to edit the various style and class files that make up the UC Exam class. If you are new to LaTeX or the UC Exam class then you should read the 'The Not So Short Guide to LaTeX2e' or 'A User's Guide to the UC Exam Class' respectively.

1 Introduction

The UC Exam class is made up of a single class file (`ucexam.cls`, Section 2) and three style-files: one for the title-page (`ucexamtitle.sty`, Section 3), one for the cover-sheet (`ucexamcoversheet.sty`, Section 4), and one for the actual questions (`ucexamquestion.sty`, Section 5). Each class and style-file has a variant associated with it that is used to typeset question and answer exams, where answers are written in the space provided below each question, rather than in a separate answer book.

The UC Exam class is split into separate files for easier maintenance, and so the individual components could be imported separately if so desired. Each of the four files is discussed in the following sections.

1.1 Nomenclature

Macros that effect the internal state of the package and are intended to be called by the exam authors start with `\UC`, while macros that typeset text do not have any prefix. However, private macros start with `\uc@`; the rest of the macro name is written using `CamelCase`. All counters, lengths and boolean values that are intended to be altered by normal users have a setter macro associated with them.

2 UC Exam Class File

The class file `ucexam.cls` is the simplest of the four main files. Its main job is to import the style files that do most of the work (Section 3–5). However, two other tasks are performed by the class file.

First, the `\UCPages` macro, which is used by the title-page (Section 3), is renewed. The title-page has no way of knowing how many pages there are in the exam, but the overall class file does by getting the page-reference of the label `uc@ExamEndOfPaper` that is placed at the end of the paper.

The class file also defines the `Exam` environment. This environment does three things:

1. Displays the title-page,

2. Displays the cover-sheet, and
3. Checks to see if the total number of marks for the exam matches the sub total that is allocated to the individual questions.

The first part of `Exam` sets the `uc@ExamTotal` total counter to hold the total number of marks in the exam, before generating the title page and cover sheet. The final part of `Exam` adds the `uc@ExamEndOfPaper` label (used in calculating the length of the document) and then checks to see if the total number of marks is the same as the sum of the marks assigned to the questions. If the two numbers do not agree a messages is displayed.

3 UC Exam Title

The UC Exam Title style-file (`ucexamtitle.sty`) creates the University-standard title page. The title page is defined in the `UCExamTitle` macro. It is responsible for the layout of the page and calls a number of internal macros to create the text associated with each part of the title.

UCUniversityNameText The name of the University.

UCDepartmentNameText The name of the department.

UCSemester The current 'semester'. This macro uses the current month to automatically determine if "Mid Year" or "End of Year" is generated.

UCAssessmentType The type of assessment, as a *plural.* It is set to the text "Examinations" by default.

UCPaperCode The course-codes of the paper. By default it is set to "Cosc666", but this is meant to be wrong. Any other papers should appear as part of a comma separated list.

UCPaperTitle The name of the paper. By default it is set to "Introduction to Doom Programming", but this is meant to be wrong. A paper should have only one title.

UCExamTime The time allowed to complete the assessment item, as a string. By default it is set to "THREE hours".

UCPages The number of pages in the assessment item. This is normally set by `ucexam.cls` (Section 2).

UCExamTitle Typesets the title page.

3.1 UC Exam Answer Title

The `ucexamanswertitle.sty` style provides the macros for typesetting the title page for the question and answer exams, which include spaces for the student to write his or her answers as well as the questions.

The style starts by importing ucexamtitle and then declares the following macros.

uc@CandidateName Formats the space that allows the student to write his or her names.

uc@CandidateNumber Formats the space that allows the student to write his or her student-ID number. This macro calls `uc@idBoxes`, which in turn calls `uc@idBox`.

uc@DoNotWrite Typesets the text “Do not write in this space”, which is used on the title page in areas that the candidate should not write on.

ecaption, listofexamples, l@example Implements a contents-like structure that typesets the boxes which allow the marker to write the marks the student receives for each question.

uc@MarksTotal Formats the box that has the total marks given for the assessment item written in it.

UCExamTitle Overwrites the original macro (found in `ucexamtitle`) in order to typeset the title page with the new information.

4 UC Exam Coversheet

The coversheet is generated by the `\UCExamCoverSheet` macro, and defined in `ucexamcoversheet.sty`. It is the `\UCExamCoverSheet` macro that most hackers would want to renew in order to provide their own instructions to students in the exams. The macro defines an itemised list, with an item for each instruction.

The default instructions were based on those given in the end of year Cosc326 exam in 2004. Three of the instructions are configurable, as they are often changed by the assessment authors.

uc@Calculators Whether calculators are allowed in the assessment item. This is implemented by a single `\ifthenelse` macro, that uses the value of the `UCCalculators` boolean to make the decision.

uc@BookText Whether the assessment item is open or closed. This macro consists of two nested `\ifthenelse` macros.

- If the value of `\UCBook` is `open` then the text for an open-book assessment item is generated.
- If the value of `\UCBook` is `closed` then the text for a closed-book assessment item is generated.
- If `\UCBook` has any other value, then that macro is used to generate the text.

This behaviour provides the assessment writer with flexibility, but covers the two most common situations.

uc@AnswerbookText Whether a separate answer book should be used. The text generated by this macro depends on the value of the boolean `UCAnswerbook`. If it is `true` then `\uc@AnswerbookText` will generate the instructions for using a separate answer book, otherwise the student will be instructed to use the spaces provided in the question book. Normally the author of an assessment item does not need to call this macro as it is called by `ucexam.cls` and `ucexamqa.cls` as appropriate.

The final macro defined in `ucexamcoversheet.sty` is `\uc@Total`, which prints out the total number of marks in the assessment item. The macro `\UCTotal` alters the value of the `uc@TotalMarksCoversheet` counter that is used by `\uc@Total`. Normally `\UCTotal` is called from the class file (`ucexam.cls`), so the author of the assessment item should not need to call it.

4.1 UC Exam Answer Coversheet

The UC Exam Answer Coversheet (`ucexamanswercoversheet.sty`) is almost the same as UC Exam Coversheet, except *all* the instructions are given to the candidates, not just the ones specific to the item of assessment. Normally students are given a list of generic instructions for exams in the post. However with a UC Question and Answer Exam, the answer book is combined with questions so we have to replicate the generic instructions.

The extra instructions are included in the renewed `UCExamCoverSheet` macro, which contains an itemised list of instructions.

5 UC Exam Questions

The `ucexamquestion.sty` file is the most complex in the entire UC Exam system, despite the fact it provides only one environment (Section 5.1) and one macro (Section 5.2).

5.1 The Questions Environment

The `Questions` environment typesets the questions. It is based on the standard `enumerate` environment, but adds functionality to check that the marks assigned to each question add up correctly.

The *begin* section of the environment increments the `uc@QuestionDepth` counter, which is used to keep track of the nesting of the questions, as this alters the behaviour of most of the other macros in the package. After the increment, the enumeration is started.

The *end* section of the environments finishes the enumeration, and then introduces a familiar idiom in this package: a series of `\ifthenelse` macros that branch on the value of the `uc@QuestionDepth` counter.

- If the value of `uc@QuestionDepth` is `1` then we have ended the top-level questions, so call the `\UCEndOfPaper` macro to finish the assessment item. By default this generates the text “End of Paper” in large centered text.
- If the value of `uc@QuestionDepth` is `2` then we have ended the subquestions, so call `\uc@CheckMarksEndII` to determine if the marks add up correctly (Section 5.3).
- If the value of `uc@QuestionDepth` is `3` then we have ended the sub-subquestions, so call `\uc@CheckMarksEndIII` to determine if the marks add up correctly (Section 5.3).

Questions can be nested only three-deep (questions, subquestions, and sub-subquestion) as this is the maximum depth supported by the LaTeX enumerate environment.

5.2 The Question Macro

The `\Question` macro takes one argument: the marks allocated to the question, which defaults to `0`. The macro is based on the standard `\item` macro, but like the *end* section of the `Question` environment (Section 5.1) it has three `\ifthenelse` macros to carry out different actions depending on the value of the `uc@QuestionDepth`.

	Subquestions	Sub-Subquestions
Start	`\uc@CheckMarksStartII`	`\uc@CheckMarksStartIII`
End	`\uc@CheckMarksEndII`	`\uc@CheckMarksEndIII`

Table 1: The macros used to check marks in `ucexamquestion.sty`. The macros that check the marks at the start of the question are called from the `Question` macro, while the `Questions` environment calls the macros to check the marks at the end.

- If the value of `uc@QuestionDepth` is `1` then add the argument to the `uc@TotalMarks` counter, set the value of the `uc@QuestionMarks` to the value of the argument, and set the value of `uc@SubQuestionMarks` to `0`.
- If the value of `uc@QuestionDepth` is `2` (a subquestion) then check the argument by calling `\uc@CheckMarksStartII` (Section 5.3). Then set the `uc@SubQuestionMarks` counter to the value of the argument, and set `uc@SubSubQuestionMarks` to `0`.
- Finally, if the value of `uc@QuestionDepth` is `3` (a sub-subquestion) then check the argument by calling `\uc@CheckMarksStartIII` (Section 5.3). Then set the `uc@SubSubQuestionMarks` counter to the value of the argument.

After the checking has been done, the `\UCQuestionItemText` macro is called. This macro typesets the text for the number of marks given to the question, which is different depending on the depth of the question.

5.3 Mark Checking

There are four macros involved with mark-checking (Table 1). The four macros are needed because there are different counters involved at the different levels, and what is checked differs between the start and end of the question.

At the start of the question the number of marks allocated to the question (the argument to the two macros) is checked to confirm that it is not greater than the number of marks allocated to the parent-question, which is stored in either the `uc@QuestionMarks` or `uc@SubQuestionMarks` counters depending if it is a subquestion or a sub-subquestion. If the check fails then a warning message is displayed to `stdout` (Section 5.3.1).

At the end of each block of questions, the `Questions` environment calls the macros to check the marks at the end. These two macros check that the total number of marks for all the subquestions or the sub-subquestions equals the marks allocated to the higher-level question. (The marks for the entire paper is checked by the `exam` environment in `ucexam.cls`, Section 2.)

5.3.1 Warning Messages

All warning messages sent to `stdout` start with the text

```
****EXAM:
```

so they can be easily distinguished from the rest of the LaTeX output. To divide each warning message the text

```
****EXAM:␣W␣A␣R␣N␣I␣N␣G
```

is sent to `stdout`.

5.4 Question and Answer Macros

The Question and Answer Macros (found in `ucexamanswerquestion.sty`) are used to typeset exams that contain a combination of questions and spaces for answers.

The style-file begins by importing `ucexamquestion.sty`, and then renews the `Question` macro. The macro is essentially the same as previously defined `Question`, but now calls the `ecaption` macro (defined in the UC Exam Answer Title style, Section 3.1) so the table of marks can be written on the front of the exam.

The macros for formatting the spaces in which answers are written are relatively simple.

AnswerLine Draws a series of ruled lines that students use as a writing guide. The gap between each line is specified by `uc@AnswerLineHeight`.

Answer Creates a box in which a series of ruled lines are written. The ruled lines are created by calling `AnswerLine`.

AnswerBox A blank box in which students can draw.

AnswerBlock A box in which a series of standard LaTeX, `AnswerLine` and `TextLine` macros appear. This allows answers to contain some pre-written text.

TextLine A macro that allows a mixture of text (supplied as an argument) and a line that the candidate can write on (drawn by `AnswerLine`).