61c-examst

v0.1.0 2025-02-03 MIT-0

A rewrite of the exam template used by CS61C at UC Berkeley in Typst.

Andrew Liu
liu.andrew@berkeley.edu

This package is used to typeset exams for the course CS61C at UC Berkeley. It is inspired by our previous exam template written in LaTeX, and contains both a template for exams, as well as various utility functions for creating exams.

Table of Contents

Part I

Quickstart

You can quickstart using the template with `typst init`:

```
typst init "@preview/cs61c-examst" exam
```

This will create the following project structure:

```
exam
├── questions
│   └── 01-sample.typ
└── main.typ
```

While you are able to write the entire exam in `main.typ`, I recommend that you write each question in its own file in the `questions` directory, and then import each question separately in `main.typ`.

Fill in each of the arguments in the

Part II

Usage

To start, import the package as follows:

```
#import "@preview/cs61c-examst:0.1.0": *
```

II.1 Template

The template for the exam is initialized by applying a show rule with the `exam` command:

```
#exam(
  ⟨class⟩,
  ⟨instructors⟩,
  ⟨semester⟩,
  ⟨exam⟩,
  ⟨time⟩,
  ⟨print-answers⟩: false,
  ⟨coverpage⟩: none,
  ⟨header⟩,
  ⟨footer⟩,
  ⟨last-edited⟩
)[body]
```

Argument

`⟨class⟩` `content` | `string`

The title of the class. This will be displayed on the cover page, as well as in the default footer.

Argument

`⟨instructors⟩` `array` | `string` | `content`

Either an array of strings or contents, or a single string / content representing the instructors of the class. This will be displayed on the cover page.

Argument

`⟨semester⟩` `string` | `content`

The name of the semester. This will be displayed on the cover page, as well as in the default footer.

Argument

`⟨exam⟩` `string` | `content`

The name of the exam. This will be displayed on the cover page, as well as in the default footer.

Argument

⟨time⟩ `string` | `content`

The amount of time allotted for the exam. This will be displayed on the cover page.

Argument

⟨print-answers⟩ `boolean`

Whether or not to print the answers to the exam. Setting this to `true` will generate the solutions to the exam, and setting this to `false` will generate a blank exam intended for printing.

Argument

⟨coverpage⟩ `string` | `function`

Either a filepath to a coverpage to use instead of the default provided with the exam template, or a function that takes in the arguments provided to `exam` (except `coverpage`) and returns content.

If not provided, the default coverpage of the template will be used.

Argument

⟨header⟩ `function` | `content`

Either a function that takes in `class, semester, exam` and returns content, or content to be shown as the header for the entire exam.

If not provided, the default header of the template will be used.

Argument

⟨footer⟩ `function` | `content`

Either a function that takes in `class, semester, exam` and returns content, or content to be shown as the footer for the entire exam.

If not provided, the default footer of the template will be used.

Argument

⟨last-edited⟩ `datetime`

The last time the exam was edited. This will be displayed only on the solutions manual. Defaults to today's date.

Part III

Utility Functions

In addition to templating the exam, this package also provides a number of utility functions for typesetting exams.

III.1 Titled Questions

The default question unit is the titled question, which displays at the top of the page with the point total for that question. In the template, top level headers automatically denote titled questions, and for headings within questions, subheadings should be used.

Titled questions can also be created using the `#titled-question` command:

```
#titled-question(⟨title⟩)
```

Argument

`⟨title⟩` `string` | `content`

The title of the question.

III.2 Parts

Within questions, subparts are automatically numbered, and the point total for each part is displayed next to the part. Subparts can be created using the `#part` command.

```
#part(⟨points⟩: none, ⟨noprompt⟩: false)
  [body]
```

The `points` argument specifies the number of points for the part, and the the `noprompt` argument is used for questions without prompts, e.g. one question with multiple multiple choice answers spread across different parts.

Points indicated here are automatically added to the point table on the front page, as well as the points for the titled question the part is under.

III.3 Question Types

Currently, multiple choice questions, free response questions with an answer blank, and coding questions are supported with the template.

The implementation of these rely on other functions in the package, and the interface is designed to be easy to use if you decide to implement your own form of question format.

III.3.1 Multiple Choice Questions

Multiple choice questions can be created using the `#multi-choice` or `#multi-select` commands:

```
#multi-choice(⟨cols⟩: 1, ⟨none-above⟩: false, ..⟨body⟩)
```

Argument
⟨cols⟩ `integer`

Argument
⟨none-above⟩ `boolean`

Argument
⟨body⟩ `any`

```
#multi-choice(cols: 1,
  [Answer 1],
  correct()[Answer 2],
  [Answer 3],
  [Answer 4],
)
```

◯ Answer 1
◯ Answer 2
◯ Answer 3
◯ Answer 4

When printing answers is enabled, the correct answer will be automatically filled in.

```
#print-answers.update(it => true)
#multi-choice(cols: 1,
  [Answer 1],
  correct()[Answer 2],
  [Answer 3],
  [Answer 4],
)
```

◯ Answer 1
● Answer 2
◯ Answer 3
◯ Answer 4

```
#multi-select((cols): 1, (none-above): false, ..(body))
```

III.3.2 Free Reponse Blanks

```
#answer-box(
  ((width: 50%)),
  ((height: 36pt)),
  (fontsize): auto,
  (pre): "",
  (post): ""
)[answer]
```

```
#answer-box()[asdf]
```

asdf

III.3.3 Code Blocks

III.3.4 Specifications

III.4 Solutions

III.5 Other Functions

III.5.1 Adding points outside of parts

III.5.2 Blankpages

III.5.3 Selectively hiding and showing content

III.5.4 Getting contextual information

III.5.5 TODOs

Part IV

Index

A

E

M

P

T