Problem Set Template

Due on 08/01/2019 at 9pm

Tadej Svetina[1]

Problem 1 (Introduction)

This is a sample document using the custom `pset` class (with the `macros` package). Just like the `notes` class, its aim is to cut down on boilerplate and provide nice and useful commands for commonly used things. In fact, the `pset` package shares all the code, apart from the theorem environments (which are not needed in problem sets) with the `notes` class[2].

That's why I will focus here on the new features that the `pset` class implements: **custom title**, **problems** and **optional solutions**. As before, the use of this class is very simple:

```
\documentclass{pset}
```

Problem 2 (Class options)

So as mentioned before, all the options are the same as for the `notes` class, except that there is no `nofancy` option. There are a few `pset` specific options:

- `nosolutions`: If this option is passed, the solutions will not be shown.

- `nobox`: If this option is passed, the solutions, if they are shown, will not be boxed.

- `inline`: If this option is passed, the problem title will be typeset inline-style (as opposed to section-style, which is default).

As in the `notes` class, all the other options get passed to the `article` class.

Problem 3 (Custom title)

The title is designed so that it conveys all the needed information about the problem set, while occupying as little space as possible. Apart from the standard `title`, `author` and `date`, the title here can also contain a `course` and a `term`. Otherwise the title creation is the same as with the normal title. For example, the title for this document was created with

```
\title{Problem Set Template}
\author{Tadej Svetina\thanks{\texttt{tadej@mit.edu}}}
\course{Template Design}
\term{Summer 2019}
\date{Due on 08/01/2019 at 9pm}

\begin{document}
```

---

[1] tadej@mit.edu
[2] In the future I should probably create a common base class.

```
\maketitle
```

Problem 4 (Problems)

The class defines a `\problem` command to show the problem title. This can be shown section-style (as in this document) or paragraph-style (if `inline` option was set, example is shown bellow). The command syntax is as follows:

```
\problem[label]{name}
```

Here `label` is used for referencing, so for example if the label is set to `ctitle` (as it is for this problem), then we can reference the problem with `\ref{pr:ctitle}`. This would produce 4.

The `name` argument simply defines a name to be used for the problem, if left empty only the counter will be shown. The number of the problem can not be controlled by this command, however it is controlled by the `problem` counter, so the usual counter manipulations apply.

Finally, here is how an inline version of this problem would look like[3]:

Problem 4 (Problems)   The class defines a `\problem` command to show the problem title. This can ...

Problem 5 (Optional solutions)

The last piece of the puzzle are the problem set solutions. They can be made to disappear – this is very useful, for example, if you are a TA, and want to avoid having to create two copies (one with solutions one without) of the problem set. You can first create the problem set with the solutions, set the `nosolutions` option to hide them and distribute this copy, and once the problem set is past the due date, remove this option and upload the full document.

By default, the solutions appear inside a box, to make them easily visually distinguishable. If you want to remove the box, pass the `nobox` option to the document class.

Here's how solutions look like:

Solutions. Does this really solve anything?

and here is how this example was produced

```
\begin{solutions}
    Does this really solve anything?
\end{solutions}
```

Finally, here's how the solutions would have looked like with the `nobox` option:

Solutions. Does this really solve anything?

---

[3]I just use `\paragraph` for this demonstration