

The csassignments class

Alexander Bartolomey

Contents

1 Introduction	2
2 Getting Started	2
3 Initial Code	3
4 Required Packages	3
5 Translations	5
6 Grouping Exercises	5
7 Document Title	8
8 Math Helpers	10

1 Introduction

This class file is designed for assignments of university courses, specifically in the field of computer science, but can easily be adapted to other fields. It provides macros for counting exercise points, adding a grading table, a helpful document title block including several non-standard information, and page headers with compressed author details.

It supports German and English language by default, by requirement of the author's university.

Additional to the layout utilities, the class provides several commonly used macros for computer science topics, namely several mathematical operators, mathmode utilities and special environments for proofs and theorem. Those are a condensed version of macros defined in <https://github.com/zoomoid/AlphabetClasses>.

2 Getting Started

Use the class like any default document class

```
\documentclass [<option>]{csassignments}
```

```

\course{}
\group{}
\due{}

\member{}{}{}
% ...

\begin{document}
\maketitle
\gradingtable

\exercise[<Exercise Points>]{<Exercise Title>}
\subexercise{}
\end{document}

```

Because it inherits from `article`, you can pass down any options that `article` understands.

3 Initial Code

First, we define the document class and set some initial lengths.

```

1 \DeclareOption*{\PassOptionsToClass{\CurrentOption}{article}}
2 \ProcessOptions\relax
3 \LoadClass[a4paper,11pt]{article}
4 \makeatletter
5 \setlength{\parindent}{0pt}
6 \setlength{\parskip}{0.6em}

```

4 Required Packages

Notably, the class introduces several dependencies. Here's a list of the packages required by `csassignments`:

geometry CTAN:macros/latex/contrib/geometry

inputenc CTAN:macros/latex/base

babel CTAN:macros/latex/required/babel/base

titlesec CTAN:macros/latex/contrib/titlesec

enumitem CTAN:macros/latex/contrib/enumitem

tocloft CTAN:macros/latex/contrib/tocloft

float CTAN:macros/latex/contrib/float

ifthen CTAN:macros/latex/base

translations CTAN:macros/latex/contrib/translations

csquotes CTAN:macros/latex/contrib/csquotes
fancyhdr CTAN:macros/latex/contrib/fancyhdr
microtype CTAN:macros/latex/contrib/microtype
stmaryrd CTAN:fonts/stmaryrd
graphicx CTAN:macros/latex/required/graphicx
pdfpages CTAN:macros/latex/contrib/pdfpages
hyperref CTAN:macros/latex/contrib/hyperref
tikz CTAN:graphics/pgf/base
microtype CTAN:macros/latex/contrib/microtype
amsmath CTAN:macros/latex/required/amsmath
amsthm CTAN:macros/latex/required/amscs
amssymb CTAN:macros/latex/required/amscs
mathtools CTAN:macros/latex/contrib/mathtools
totcount CTAN:macros/latex/contrib/totcount
changepage CTAN:macros/latex/contrib/changepage
etoolbox CTAN:macros/latex/contrib/etoolbox
environ CTAN:macros/latex/contrib/environ
multicol CTAN:macros/latex/required/tools
tgpagella CTAN:fonts/tex-gyre
suffix CTAN:macros/latex/contrib/bigfoot

```

7 \RequirePackage[left=2.5cm,right=2.5cm,top=2cm,bottom=2cm,
8  includeheadfoot]{geometry}
9 \RequirePackage[utf8]{inputenc}
10 \RequirePackage[english]{babel}
11 \RequirePackage{titlesec}
12 \RequirePackage{enumitem}
13 \RequirePackage{graphicx}
14 \RequirePackage{tocloft}
15 \RequirePackage{float}
16 \RequirePackage{ifthen}
17 \RequirePackage{translations}
18 \RequirePackage{csquotes}
19 \RequirePackage{fancyhdr}
20 \RequirePackage{microtype}
21 \RequirePackage{stmaryrd}
22 \RequirePackage{pdfpages}
23 \RequirePackage[hidelinks, unicode]{hyperref}
24 \RequirePackage{tikz}

```

```

25 \RequirePackage{microtype}
26 \RequirePackage[fleqn]{amsmath}
27 \RequirePackage{amsthm}
28 \RequirePackage{amsymb}
29 \RequirePackage{mathtools}
30 \RequirePackage{totcount}
31 \RequirePackage{changepage}
32 \RequirePackage{etoolbox}
33 \RequirePackage{environ}
34 \RequirePackage{multicol}
35 \RequirePackage{tgpagella}
36 \RequirePackage{suffix}
37 \usetikzlibrary{arrows,calc}

```

5 Translations

csassignments comes with several predefined prefixes that are localized in both English and German:

```

38 \DeclareLanguage{english}
39 \DeclareLanguage{german}
40 \select@language{english}
41 \ProvideTranslation{german}{prefixAuthor}{Eingereicht von}
42 \ProvideTranslation{english}{prefixAuthor}{Handed in by}
43 \ProvideTranslationFallback{prefixAuthor}{Handed in by}
44 \ProvideTranslation{german}{prefixDate}{am}
45 \ProvideTranslation{english}{prefixDate}{on}
46 \ProvideTranslationFallback{prefixDate}{on}
47 \ProvideTranslation{ngerman}{exercise}{Aufgabe}
48 \ProvideTranslation{english}{exercise}{Exercise}
49 \ProvideTranslationFallback{exercise}{Exercise}
50 \ProvideTranslation{ngerman}{exerciseAbbrev}{Aufg.}
51 \ProvideTranslation{english}{exerciseAbbrev}{Ex.}
52 \ProvideTranslationFallback{exerciseAbbrev}{Ex.}
53 \ProvideTranslation{ngerman}{subexercise}{Teilaufgabe}
54 \ProvideTranslation{english}{subexercise}{Subexercise}
55 \ProvideTranslationFallback{subexercise}{Subexercise}
56 \ProvideTranslation{ngerman}{points}{Punkte}
57 \ProvideTranslation{english}{points}{Points}
58 \ProvideTranslationFallback{points}{Points}
59 \ProvideTranslation{ngerman}{sheet}{Abgabe}
60 \ProvideTranslation{english}{sheet}{Exercise Sheet}
61 \ProvideTranslationFallback{sheet}{Exercise Sheet}
62 \ProvideTranslation{ngerman}{studentId}{Matr.Nr.}
63 \ProvideTranslation{english}{studentId}{Student ID}
64 \ProvideTranslationFallback{studentId}{Student ID}

```

6 Grouping Exercises

To support the grading table, we need to initialize several counters for both counting up the awarded points of exercises but also for labels of exercises and subexercises.

```

65 \newtotcounter{exercisenum}
66 \newtotcounter{subexercisenum}
67 \newtotcounter{totalpoints}
68 \edef\exerciseLabelGradingTable{}
69 \edef\gradingTableExercisePoints{}
70 \edef\PTableA{}
71 \edef\PTableB{}
72 \edef\PTHead{}
73 \newcommand{\@addPT}[2]{
74   \edef\gradingTableExerciseLabel{\GetTranslation{exerciseAbbrev} #1 &}
75   \edef\gradingTableExercisePoints{\quad / $#2$ &}
76   \xappto\PTableA{\gradingTableExerciseLabel}
77   \xappto\PTableB{\gradingTableExercisePoints}
78   \xappto\PTHead{c | }
79 }

```

`\gradingtable` You can toggle the displaying of the grading table by adding `\gradingtable` at some point in your document before the first exercise. It requires the compiler to run twice, in order expand the control sequence created on the first run, where exercise labels and points were appended in order to be combined in a subsequent run.

```

80 \newcommand{\gradingtable}{%
81   \ifdef{\PTHeadSaved}{%
82     {\Large\bfseries Grading Table}\%
83     \begin{table}[h!]
84       \renewcommand\arraystretch{2}%
85       \begin{tabular}{\expandonce{\PTHeadSaved} c}
86         \PTableASaved {\Large$\Sigma$}\ \hline
87         \PTableBSaved \quad / $\totalpointsSaved$
88       \end{tabular}
89       \renewcommand\arraystretch{1}%
90     \end{table}
91   }{\texttt{Compile again}\ClassWarning{csassignments}{Compile again to %
92     properly display grading table}}
93 }

```

`\exercise` On top-level, assignments consist of one or more exercise. The `\exercise[exercise points]{exercise title}` macro provides a numbered section title for your assignment. The mandatory argument should contain the exercise's title which is then included in the section title, or be left blank, in which case nothing is appended to the exercise title. The optional argument contains the points rewarded for the exercise in total.

Exercise points are added up in the first run of `pdflatex` (or your TeX backend of choice) and written to the auxiliary file. On the second run, the grading table is added to the top of the document, by inclusion of the points stored in the first run. **This is the reason `csassignments` requires two or more runs when used with the grading table.**

If you do not specify points, that is, leave the optional argument field empty, the awarded points for the exercise will be set to zero, but the exercise will not be included in the grading table.

If you do not specify a title, no empty parentheses are printed.

Using the starred version of the exercise command will allow you to explicitly exclude exercise from the grading table (similar to `\section*{section title}`)

not being included in the table of contents). Note that you can set points on explicitly excluded exercises. The exercise will not count towards the total number of points, but the points for the exercise will still be included in the right column of the exercise's title line. This might be particularly helpful for modelling bonus exercises.

```

94 \newcommand{\@pointsPerExercise}[1]{
95   \ifstrempy{#1}{{\quad\large\bfseries/#1 \GetTranslation{points}}}
96 }
97 \newcommand{\@exerciseLabel}[1]{
98   \quad\ifstrempy{#1}{{\bfseries(#1)}}
99 }
100 \newcommand\exercise[2][]{
101   \penalty150
102   \ifblank{#1}{\def\points{0}}{\def\points{#1}}
103   \stepcounter{exercisenum}
104   \setcounter{subexercisenum}{0}
105   \par
106   \vspace{1.5\baselineskip}
107   {
108     {\Large\bfseries\GetTranslation{exercise}\ \arabic{exercisenum}}
109     \@exerciseLabel{#2}
110     \@exerciseRules
111     \notblank{#1}{\@pointsPerExercise{#1}}{}
112   }
113   \par
114   \notblank{#1}{
115     \addtocounter{totalpoints}{\points}
116     \@addPT{\arabic{exercisenum}}{\points}
117   }{}
118 }
119 \WithSuffix\newcommand\exercise*[2][]{
120   \penalty150
121   \ifblank{#1}{\def\points{0}}{\def\points{#1}}
122   \stepcounter{exercisenum}
123   \setcounter{subexercisenum}{0}
124   \par
125   \vspace{1.5\baselineskip}
126   {
127     {\Large\bfseries\GetTranslation{exercise}\ \arabic{exercisenum}}
128     \@exerciseLabel{#2}
129     \@exerciseRules
130     \notblank{#1}{\@pointsPerExercise{#1}}{}
131   }
132   \par
133   \addtocounter{totalpoints}{\points}
134 }
135 \AtEndDocument{
136   \makeatletter
137   \immediate\write\@mainaux{\string\gdef\string\PTableASaved{\PTableA}}
138   \immediate\write\@mainaux{\string\gdef\string\PTableBSaved{\PTableB}}
139   \immediate\write\@mainaux{\string\gdef\string\PTHeadSaved{\PTHead}}
140   \immediate\write\@mainaux{\string\gdef\string\totalpointsSaved%
141     {\arabic{totalpoints}}}

```

```
142 \makeatother
143 }
```

`\subexercise` Subexercises are the subsection to section equivalent for exercises. In fact, their counters behave the same way as subsection counters do. `\subexercise[(Subexercise title)]` will make a new subexercise under the current exercise. Note that point counting for subexercises is not supported. Rather, points are only awarded on the exercise level. Therefore, subexercises are rather semantical for the exercise's reading flow.

```
144 \newcommand{\subexercise}[1] []{
145   \penalty150
146   \stepcounter{subexercisenum}
147   \par
148   \vspace{0.5\baselineskip}
149   {
150     {\bfseries\GetTranslation{subexercise}\ (\alph{subexercisenum})}
151     \quad\ifstrempy{#1}{}{( #1)}
152     \@subexerciseRules
153   }
154   \par
155 }
```

```
156 \def\Vhrulefill{\leavevmode\leaders\hrule height 0.7ex depth
157 \dimexpr0.4pt-0.7ex\hfill\kern0pt}
```

`\exerciseRules` To add line-spanning horizontal rules between exercises and subexercises, respectively, for better visual separation, add `\exerciseRules` add some point in the document before the first exercise.

```
158 \newcommand{\exerciseRules}{
159   \renewcommand{\@exerciseRules}{\hspace{1em}\Vhrulefill} }
160 \newcommand{\subexerciseRules}{
161   \renewcommand{\@subexerciseRules}{\hspace{0em}\Vhrulefill} }
162 \newcommand{\@exerciseRules}{}
163 \newcommand{\@subexerciseRules}{}

```

`\noPoints` To remove points and grading entirely from the document, add `\noPoints` at some point in the document before the first exercise. This overwrites the grading table macro and defines the exercise command to not count points.

```
164 \newcommand{\nopoints}{
165   \renewcommand{\@pointsPerExercise}{}
166   \renewcommand{\gradingtable}{}
167   \renewcommand{\exercise}{\@exerciseNoPoints}
168 }
169 \newcommand{\@exerciseNoPoints}[1]{
170   \def\points{0}
171   \stepcounter{exercisenum}
172   \setcounter{subexercisenum}{0}
173   \par
174   \vspace{1.5\baselineskip}
175   { {\Large\bfseries\GetTranslation{exercise}\ \arabic{exercisenum}}
176     \@exerciseLabel{#1} \@exerciseRules }
177   \par
178 }
```

7 Document Title

The `csassignments` class offers several customizations to the default title of article. Namely, we can omit the default fields `\title`, `\author`, and `\date`, and replace them with something more fitting to an assignment:

```

179 \newcommand{\@course}{Assignment}
180 \newcommand{\@sheet}{}
181 \newcommand{\@group}{}
182 \newcommand{\@prefixId}{\GetTranslation{studentId} }
183 \newcommand{\@members}{}
184 \newcommand{\@member}[2] []%
185   {\notblank{#1}{#2, \notblank{#1}{\@prefixId{#1}}}} \\\{#2}}
186 \newcommand{\@due}{}
187 \newcommand{\@prefixAuthor}{\GetTranslation{prefixAuthor}}
188 \newcommand{\@prefixDate}{\GetTranslation{prefixDate}}

```

`\course` Course name with `\course{<Course>}`, sheet number with `\sheet{<sheet number>}`,
`\sheet` group name with `\group{<group name>}`, due date with `\due{<due date>}`, and
`\group` all members of the group with repeated `\member[<student id>]{<name>}`.

```

\due 189 \newcommand{\course}[1]{\expandafter\notblank\expandafter{#1}%
\member 190   {\def\@course{#1}}}}
191 \newcommand{\sheet}[1]{\def\@sheet{\GetTranslation{sheet} #1}}
192 \newcommand{\group}[1]{\renewcommand{\@group}{#1}}
193 \newcommand{\due}[1]{\renewcommand{\@due}{#1}}
194 \newcommand{\member}[2] []{
195   \expandafter\renewcommand\expandafter\@members\expandafter
196     {\@members\@member[#1]{#2}}
197   \expandafter\renewcommand\expandafter\pdfmembers\expandafter
198     {\pdfmembers\pdfmember[#1]{#2}}
199 }

```

`\PrefixId` Additionally, you can redefine the prefixes prepended to the individual parts of
`\PrefixAuthor` the title, for example, the default (english) prefix for the members line in the title
`\PrefixDate` block is “Handed in by”, which you can replace by using `\PrefixDate{<custom due date prefix>}`.

```

200 \newcommand{\PrefixId}[1]{\renewcommand{\@prefixId}{#1}}
201 \newcommand{\PrefixAuthor}[1]{\renewcommand{\@prefixAuthor}{#1}}
202 \newcommand{\PrefixDate}[1]{\renewcommand{\@prefixDate}{#1}}

203 \renewcommand{\and}{\}
204 \renewcommand{\maketitle}{
205   \thispagestyle{empty}
206   {\bfseries\Huge\@course}
207   \par
208   \expandafter\notblank\expandafter\@sheet{\LARGE\@sheet\}\}
209   \par
210   \expandafter\notblank\expandafter\@members{
211     {\footnotesize\bfseries\@prefixAuthor}\}
212     \hspace*{4pt}{\large{\vspace*{4pt}\begin{tabular}[t]{@{}l}
213       \@members
214       \end{tabular}}}\}
215   }{}
216   \expandafter\notblank\expandafter\@due{

```



```

217   {\footnotesize\bfseries\@prefixDate}\
218   \vspace*{4pt}{\hspace*{4pt}{\large\@due}}\
219   }{}}
220   \vskip \baselineskip
221   \pagestyle{fancy}
222 }
223 \renewcommand{\headrulewidth}{1pt}
224 \setlength\headheight{4\baselineskip}
225 \rhead{
226   \begin{tabular}{r}
227     \textbf{\@course} \
228     \sheet \
229     {\small\@due}
230   \end{tabular}
231 }
232 \lhead{
233   \small{%
234     \begin{tabular}{l}
235       \@members
236     \end{tabular}%
237   }
238   }

```

Lastly, the class handles PDF metadata creation using hyperref from the provided macros, i.e., `\course`, `\group` etc.

```

239 \newcommand{\@pdfauthorprefix}{\expandafter\notblank\expandafter%
240   {\@group}{Group \@group: }{\relax}
241 \newcommand{\pdfmembers}{}
242 \newcommand{\pdfmember}[2] [] {\expandafter\notblank\expandafter%
243   {#1}{#2, {#1};}{#2}}
244 \AtBeginDocument{\hypersetup{
245   pdftitle = {\texorpdfstring{Assignment \@sheet}{}},
246   pdfauthor = {\texorpdfstring{\@pdfauthorprefix\pdfmembers}{}},
247   pdfsubject = {\texorpdfstring{\@course}{}}
248 }}

```

8 Math Helpers

`csassignments` includes several frequently used macros for mathmode, that were, and still are, helpful with assignment work during several computer science courses. Most of them were originally created in one of the `ALPHABETCLASSES` packages, which featured a lot more definitions.

This class only contains a condensed version of them:

```

\N These macros define common number spaces and fields as double font in math-
\Z mode. Notably, \primefield makes a double-printed P, as notation for an arbi-
\R trary prime field.
\Q 249 \newcommand{\N}{\ensuremath{\mathbf{N}}}
\C 250 \newcommand{\Z}{\ensuremath{\mathbf{Z}}}
\F 251 \newcommand{\R}{\ensuremath{\mathbf{R}}}
\primefield 252 \newcommand{\Q}{\ensuremath{\mathbf{Q}}}
253 \newcommand{\C}{\ensuremath{\mathbf{C}}}

```

254 `\newcommand{\F}{\ensuremath{\mathbb{F}}}`
255 `\newcommand{\primefield}{\ensuremath{\mathbb{P}}}`

`\modring` Denoting a ring of integers modulo n is easily done in short notation with `\modring`, see \mathbb{Z}/n , which even works in regular text due to custom trailing spacing.

256 `\newcommand{\modring}[2]{\!/\!\!#2\!}`

`\derivative` The `\derivative{⟨variable⟩}` prints a typical $\frac{d}{dx}$ as a shorthand macro. Notably it is not even shorter than the explicit fracture, it is just a convenience macro created at a time when writing \LaTeX for assignments was supported by semantic macros (a thing \LaTeX clearly lacks).

257 `\newcommand{\derivative}[1]{\ensuremath{\frac{d}{d#1}}}`

`\matadd` These macros are helpful when visualizing matrix operations, namely addition
`\matmul` of rows, scalar multiplication of a row, and swapping of rows. Arguments of the
`\matswap` macros are identical to the group operation's parameter.

258 `\newcommand{\matadd}[3]{\ensuremath{\xmapsto{\mathrm{add}}_{#1,#2,#3}}}`
259 `\newcommand{\matmul}[2]{\ensuremath{\xmapsto{\mathrm{mul}}_{#1,#2}}}`
260 `\newcommand{\matswap}[2]{\ensuremath{\xmapsto{\mathrm{sw}}_{#1,#2}}}`

`\forall` To make logic quantors look a bit better, we redefine the “for all” and “exists”
`\exists` quantors with a bit more horizontal space left and right

261 `\renewcommand{\forall}{\ensuremath{\hspace{2pt}\forall\hspace{2pt}}}`
262 `\renewcommand{\exists}{\ensuremath{\hspace{2pt}\exists\hspace{2pt}}}`

`\floor` These are some commonly used mathematical functions, that is, rounding up or
`\ceil` down, and the absolute function of a value.

`\abs` 263 `\newcommand{\ceil}[1]{\ensuremath{\left\lceil #1 \right\rceil}}`
264 `\newcommand{\floor}[1]{\ensuremath{\left\lfloor #1 \right\rfloor}}`
265 `\newcommand{\abs}[1]{\ensuremath{\left| #1 \right|}}`

`\rfrac` For inline math, regular fractions tend to either stretch the line height, resulting in optical heterogeneity of multiple lines, or setting the letters in the fraction so small that they become unreadable. For small, inline fractions, `\rfrac{⟨numerator⟩}{⟨denominator⟩}` offers a less optically noisy option, by setting the numerator and denominator in a diagonal arrangement, rather than purely vertical. Here are some examples: $\frac{1}{n}$, $\frac{2n}{k}$, $\frac{100}{4}$.

266 `\newcommand{\rfrac}[2]{\!/\!_{#1}\!/_{#2}}`

`\mod` The macro defines the usual modulus function with a bit of extra space to the left and right.

267 `\renewcommand{\mod}{\ \mathrm{mod}\ }`

`\rel` To make a new, arbitrary relation, that is, a symbol for with `amsmath` ensures horizontal space left and right using `\mathrel`

268 `\newcommand{\rel}[1]{\ensuremath{\mathrel{#1}}}`

`\QED` Ending a proof the way taught at RWTH Aachen University, with a square flushed to the right of the line, can be done by simply adding `\QED`. Note however that it

is semantically more elegant to use `amsthm` with either predefined `proof` environment, or any custom environment, as these usually also handle adding a label to the block, e.g., “**Proof.**” or “**Lemma**”.

```
269 \newcommand{\QED}{\hfill(\square)}
```

```
270 \newtheoremstyle{it}{3pt}{3pt}{\itshape}{.5em}{}
```

```
271 \theoremstyle{it}
```

```
272 \makeatother
```