The fhjcommon package*

Michael E. Hammer
michael.hammer@fh-joanneum.at

November 19, 2026

Abstract

Collection of common macros for the FHJ bundle independent of a chosen document class.

Contents

1 Configuration **1**
1.1 Used Packages 1

2 Naming definitions **1**
2.1 Study program specifics 2

3 Legal declarations **3**

4 Exercises **3**
4.1 Exercise headers 3
4.2 Example section 4
4.3 LLM usage 4

1 Configuration

1.1 Used Packages

```
1 ⟨*package⟩
2 \RequirePackage{xstring}
3 \RequirePackage{ifthen}
4 \RequirePackage{calc}
```

2 Naming definitions

\FHJ One goal of this package is to provide common naming definitions which can and should be used in all documents which are created in the context of *FH JOANNEUM*. This includes the naming of the university (\FHJ), the study programs and their long and short names.

*This document corresponds to fhjcommon v4.3e, dated 2026-11-19.

2.1 Study program specifics

\FHJProgram It is assumed that the documents which are generated are created in the context of a given study program. To set the study program for the given document use the macro as following:

\FHJProgram{⟨*code*⟩}

This updates what \FHJProgramLong and \FHJProgramShort expand to – the long and short name registered for {⟨*code*⟩}. Passing an unregistered {⟨*code*⟩} raises a package error listing the currently known codes. \FHJProgram also accepts ITP, which is not a study program but refers to the institute itself – see the remark at the end of this subsection.

\FHJProgramLong \FHJProgramShort and \FHJProgramLong[⟨*code*⟩] and \FHJProgramShort[⟨*code*⟩] expand to the long and short name of the currently set study program. Given the optional [⟨*code*⟩], they first call \FHJProgram{⟨*code*⟩} – switching the study program for the rest of the document – and then expand to its long/short name; this is convenient for setting and using the study program name in a single place, without a separate \FHJProgram call beforehand.

\FHJDefineProgram Study programs are registered in a small table, keyed by a {⟨*code*⟩} that stays stable for a program's lineage while it may carry a year suffix marking when a given name *first became official* (e.g. IMS-26) – this is not a student cohort year. New codes are registered with

\FHJDefineProgram{⟨*code*⟩}{⟨*short name*⟩}{⟨*long name*⟩}

{⟨*code*⟩} may contain letters, digits and hyphens. The {⟨*short name*⟩} is stored separately from {⟨*code*⟩} on purpose – a code's year suffix should not leak into the displayed short name, so e.g. both DAT-22 and DAT-25 use the short name DAT. For example \FHJDefineProgram{AIM-27}{AIM}{Applied Computer Science} registers AIM-27 so that \FHJProgram{AIM-27} afterwards sets \FHJProgramShort to "AIM" and \FHJProgramLong to "Applied Computer Science". This is the intended extension point for future study programs: maintainers or downstream users can call \FHJDefineProgram for a new program without editing \FHJProgram itself, as long as the call happens before \FHJProgram is used. The code ITP is reserved for the institute (see below) and cannot be registered this way.

Registered {⟨*code*⟩} values are currently:

```
\FHJDefineProgram{DAT-22}{DAT}{Data Science and Artificial Intelligence}
\FHJDefineProgram{DAT-25}{DAT}{Data Science and Artificial Intelligence}
\FHJDefineProgram{AIM-22}{AIM}{Business Informatics}
\FHJDefineProgram{AIM-27}{AIM}{Applied Computer Science}
\FHJDefineProgram{IMS-24}{IMS}{IT and Mobile Security}
\FHJDefineProgram{IMS-26}{IMS}{Cybersecurity and Ethical Hacking}
```

ITP is not a study program but refers to the institute itself: calling \FHJProgram{ITP} (or \FHJProgramShort[⟨*ITP*⟩]/\FHJProgramLong[⟨*ITP*⟩]) sets \FHJProgramShort to \FHJInstitute – tracking its *current* value rather than a snapshot – and \FHJProgramLong to a fixed "Institute for Information Technology+" (both handled inside \FHJProgram above).

3 Legal declarations

Further important macros are the legal declarations which are required to be included in the master thesis. These declarations ensure compliance with the legal requirements of FH JOANNEUM.

\FHJMakeDeclaration

Normally these declarations should be placed on a single page – therefore we run a `\cleardoublepage` before and after the declaration. The macro `\FHJDeclaration` is also taking care of the language and will automatically switch to the appropriate language based on the current language setting of the document. We currently support *ngerman* and *english* as languages. The declaration text itself is defined in both German and English and could be typeset with `\FHJDeclarationDE` or `\FHJDeclarationEN` respectively without any page clearing.

\FHJDeclarationDE
\FHJDeclarationEN

4 Exercises

The class provides a set of macros to define exercises or exams and according examples for lectures. To be able to create the header you will need to redefine the following macros.

\FHJClassAbbr
\FHJClassTerm
\FHJLecturer

To simplify the generation of the header for exercises we provide a set of macros which can be redefined to customize the appearance of the header.

```
\newcommand{\FHJClassAbbr}{RUL}
\newcommand{\FHJClassTerm}{WS 2025-26}
\newcommand{\FHJLecturer}{Lisa Musterfrau}
```

4.1 Exercise headers

\FHJExHeader

The header is actually generated with the macro `\FHJExHeader[`⟨*type*⟩`]{`⟨*id*⟩`}{`⟨*topic*⟩`}`: an optional exercise type (like "Exam", "Exercise" or similar; default "Exercise"), followed by the exercise id and the topic. Note the type is the *optional* argument – it must be given in brackets. It may look like this with the above default definitions:
with `\FHJExHeader[Exercise]{1}{\LaTeX}`:

RUL (Data Science and Artificial Intelligence)
FH JOANNEUM
Lisa Musterfrau

Exercise 1: LaTeX WS 2025-26

\FHJExHeaderScheduled

Additionally we defined the macro `\FHJExHeaderScheduled[`⟨*type*⟩`]{`⟨*id*⟩`}{`⟨*topic*⟩`}{`⟨*date*⟩`}` which takes an additional argument for the date which would produce the following example:

RUL (Data Science and Artificial Intelligence)
FH JOANNEUM
Lisa Musterfrau

Exercise 1a: LaTeX WS 2025-26

Exercise 1a (LaTeX) to be solved till 2025-08-13
Remark: The Exercises have to be handed in on Moodle on the target day until 23:59.

4.2 Example section

`\FHJExExample` To generate the examples themselves you may want to use the macro `\FHJExExample[`⟨*points*⟩`]{`⟨*title*⟩`}` which will generate by using `\FHJExExample[`⟨*99*⟩`]{`⟨*An amazing challenge!*⟩`}` the following section:

Example 1a.1: An amazing challenge! (99 Points)

The section counter will be incremented automatically and the major number is always equal to the exercise id which is set by previous call to `\FHJExHeader` or `\FHJExHeaderScheduled`.

4.3 LLM usage

`\FHJExLLM` We have formulated a default statement for exercises on how to treat LLMs (Large Language Models) in the context of the FHJ. This is not an "official" statement from FHJ - it is more a guideline for students and lecturers on how to approach the topic.

Usage of LLMs: You are permitted to use LLMs (AI assisstents) to support your work on the exercises. Please note, however, that the use of LLMs is not allowed during assessments. Ensure that you thoroughly understand your solutions and can reproduce them independently without assistance from LLMs.

Change History

v4.0
General: Initial package version . . 1
v4.3
General: `\FHJProgramLong` and `\FHJProgramShort` take an optional ⟨*code*⟩ that sets the study program inline 1
Removed `\FHJDATLong`, `\FHJDATShort`, `\FHJIMSLong` and `\FHJIMSShort` 1
Study programs moved to a data table registered via `\FHJDefineProgram`; added Applied Computer Science, Business Informatics and Cybersecurity and Ethical Hacking 1
v4.3b
General: `\FHJExHeaderScheduled` now forwards the type as the optional argument to `\FHJExHeader`, so the exercise id is no longer overwritten by the type 1