

# Effective Programming Practices for Economists, Autumn 2024, Bonn

## Contents

- Hans-Martin von Gaudecker
- Course learning objectives

## Hans-Martin von Gaudecker

Welcome to the course *Effective Programming Practices for Economists*, where we aim to teach basic software engineering principles that should be useful for economists.

Software engineering skills are essential for anyone who does applied econometrics, machine learning or computational economics. They are also helpful for theorists who want to do simulations. After attending this class you will be able to do fully reproducible research projects in Python or other languages. See below for more detailed learning objectives.

The course will be taught in the flipped classroom format. This means that you work through materials **before the class** and in the actual class you apply the knowledge you gained in hands-on exercises.

A typical preparation for a class involves:

- Watch multiple short videos at your own pace
- Do multiple choice quizzes to test what you have learned
- Prepare questions for the lecture about anything that remains unclear

The typical class will then consist of:

- Q&A session
- Simple exercises where you apply the exact things you learned from videos
- More demanding exercises where you deepen your knowledge
- Work on the current assignment or discuss solutions to previous assignments

All practical exercises and assignments are solved in groups.

## Course learning objectives

Here we explain at very high level what you may expect to learn in this course. The teaching style and the work you need to do in order to get credit are designed to make reaching these objectives efficient. We have a [separate page](#) on grading, however.

We group the learning objectives into two categories:

### Methods (core concepts and abstract coding skills)

- Students master the basic techniques to make current and past versions of their research reproducible
- Students solve reasonably complex economic problems using the help of a computer
- Students are able to write correct code in an efficient manner employing modern software engineering techniques
- Students reason about the structure of economic data and handle such data efficiently

### Tools (specific programmes, which you may want to use in the long run or not)

- Students use a shell for navigating the directory system and for starting programmes
- Students routinely use *Git* for version control and collaborating with coauthors
- Students manage economic data using basic *pandas* functionality
- Students automate repetitive tasks using Python

- Students run their analysis pipelines using *pytask*
- Students find solutions to specific programming problems through sensible use of AI, search engines, and websites like *Stack Overflow*
- Students visualize data using *plotly*
- Students speed up their code using *numpy* and *numba*
- Students write automated unit tests using the *pytest* framework