

Design Tools And Laboratory Engineering - Coding

Course Number M24.0862104, WS 2025/26

Bernhard Reitingner

```

##      ## ##### ##      #####      ##### ##      ## #####
##  ##  ##  ##      ##      ##      ##  ##      ##  ##
##  ##  ##  ##      ##      ##      ##  ##      ##  ##
##  ##  ##  #####  ##      ##      ##  ##  ##  ##  #####
##  ##  ##  ##      ##      ##      ##  ##      ##  ##
##  ##  ##  ##      ##      ##      ##  ##      ##  ##
      ###  ###  #####  #####  #####      #####  ##      #####

```

ABOUT ME

- 🖥️ Coding is my passion → 30+ years of experience
- 🎓 PhD in Computer Science (TU Graz)
- 🚀 Software entrepreneur
- 🖥️ Terminal & neovim enthusiast – I value simplicity
- 🌐 Explored many languages
 - Pascal, Oberon, C/C++, Java/Groovy, C#, JavaScript, Shell, Python, Go, Lua
- 🎯 My mission here: → Help you get started with professional tools & workflows



COURSE GOALS & EXPECTED OUTCOMES

GOAL

Each student can develop, test, and share a small C++ project in a Linux environment

IN DETAIL

- Understand & use essential tools
 - Linux, shell, tmux, neovim, Git, C++17, CMake, STL, debugging, testing
- Build professional workflows
- Navigate Linux confidently
- Manage code with Git & branches
- Edit, compile, and debug C++ projects
- Apply to engineering domains

COURSE STYLE & LEARNING APPROACH

- Hands-on first → minimal slides, maximum coding
- Mini-exercises every session
- Direct application of new concepts
- Short, focused tasks (5-10 min)
- Project-oriented learning
- Start small → grow into a working project
- Engineer-relevant examples
- Collaboration & best practices
- Use Git like a professional
- Debug, test, and document code

Questions?

