Lecture X - Programming Projects

Programming with Python

Dr. Tobias Vlćek

Kühne Logistics University Hamburg - Fall 2025

Quick Recap of the last Lecture

General

Congratulations

You've learned your first steps to program with Python! 🎉

. . .

Structure

- Over the upcoming weeks you will work on a project
- You will present it in the last week of this course
- You can work in groups of up to 3 people
- Choose from a list of ideas or propose your own idea!

. . .

> 💡 Tip
>
> You have enough time to discuss different ideas in your group today. From my experience, it is a good idea to choose a project that you are really interested in.

Presentation

- Each group has 10 minutes for the presentation with 5 additional minutes for questions
- Introduce your idea and the development cycle
- Provide code examples and/or visualizations
- Comment on challenges and what you've learned

. . .

> 💡 Tip
>
> Your project does not have to be perfect! To pass, you simply have to show that you tried your best. Try, fail potentially and learn - that's the best way to improve your coding skills.

Project Ideas

Idea 1: Advanced Data Analytics Platform
- Build a complete data pipeline
- Implement automated data cleaning and validation
- Create interactive dashboards with real-time updates
- Add predictive modeling and automated reporting

. . .

Bonus: Integrate multiple data sources, add email alerts

Idea 2: Intelligent Web Scraping System
- Build a multi-source scraping pipeline
- Implement error handling and retry logic
- Store data in a database (SQLite or PostgreSQL)
- Create a monitoring dashboard with visualization
- Add automated alerts when target data changes

. . .

Bonus: Add proxy rotation, detect website structure changes

Idea 3: ML Model with Full Deployment
- Build an end-to-end ML pipeline for a real-world problem
- Implement data preprocessing, feature engineering, and model training
- Build a web interface for users to interact with the model
- Add model monitoring and performance tracking

. . .

Bonus: A/B testing, explainability dashboard

Idea 4: Real-Time Computer Vision System
- Build a real-time video processing pipeline
- Implement object detection, tracking, and analytics
- Create a web interface to view live results
- Store and analyze detection statistics over time

. . .

Bonus: Multi-camera support, alert system

Idea 5: Advanced Simulation System
- Build a complex simulation
- Implement Monte Carlo methods or agent-based modeling
- Create interactive visualizations with parameter controls
- Add sensitivity analysis and scenario comparison
- Generate automated reports with statistical analysis

. . .

Bonus: GPU acceleration, parallel processing

Idea 6: Game Development

- Create a complete game with multiple levels/stages
- Add save/load system, high scores, and user profiles
- Include sound effects, music, and polished graphics

. . .

Bonus: Procedural level generation, achievement system

Idea 7: Intelligent Automation System

- Build a robust automation framework for repetitive tasks
- Implement error detection, logging, and automatic recovery
- Create a web interface to monitor and control automation
- Add scheduling, notifications (email/Slack), and reporting
- Include configuration management for different scenarios

. . .

Bonus: OCR for screen reading, computer vision for UI detection

Idea 8: Your Idea?

- Have an idea that is not on the list?
- Make it ambitious!
- Consider: deployment, testing, user interface, data persistence
- Combine multiple technologies and concepts
- Let me know and we can discuss scope and feasibility!

Help over the upcoming weeks

Ask Questions

- In case you need help, you can always ask me!
- The next lectures are there to work on your project
- You can also write me an email at vlcek@beyondsimulations.com

> 💡 Tip
>
> I am always happy to help you with your project. There are no stupid questions!

Use of AI Tools

- Strongly encouraged to use AI tools for your project!
- Remember: AI accelerates development, but understanding is essential
- Use AI to learn patterns, debug issues, and explore solutions

. . .

> 💡 Tip
>
> Try multiple AI tools to find what works best for your workflow. Claude Code, Zed, Copilot, Cursor, OpenCode and LM Studio are all excellent choices!

Agentic Coding

Agentic coding means AI agents work autonomously on tasks

. . .

How it works:

- You describe what you want to achieve
- The AI agent plans and executes multiple steps
- It can run tests, fix errors, and iterate independently
- You review and guide the overall direction

. . .

Best for: Complex refactoring, adding test suites, documentation generation, multi-step implementations

Claude Code

Claude Code is an AI coding assistant by Anthropic

. . .

Key Features:

- Understands entire codebases, not just single files
- Can edit multiple files simultaneously
- Explains complex code patterns clearly
- Helps with debugging and refactoring

. . .

> 💡 Tip
>
> Great for: Understanding project structure, implementing features across multiple files, learning best practices

Alternative AI Coding Tools

Beyond Claude Code, many tools are available:

. . .

Popular Options:

- Open Code: Open Source alternative to Claude Code
- Cursor: AI-first code editor based on VS Code
- Zed: Lightweight editor with Claude & ChatGPT integration

. . .

Each tool has strengths, experiment to find your preference!

LM Studio - Local AI for Privacy
[LM Studio](#) runs AI models entirely on your computer

. . .

Benefits:

- Complete privacy - your code never leaves your machine
- No API costs or rate limits
- Works offline with full control over model selection

. . .

> 💡 Tip
>
> Ideal if you're working with sensitive data or want to learn without internet dependency. Requires a decent GPU for best performance.

Discuss your ideas!

How to continue?

How to continue after the presentations?
- The best way to continue learning is to keep programming in the future
- Potentially, you will continue to do so during your studies
- Coding in your Thesis is a another great way to improve
- Try to find a way to apply programming in your work
- There are many interesting topics to explore!

Advent of Code
- [Advent of Code](#) is a fun way to keep programming
- Here you can solve programming puzzles during Advent
- It is completely free and ad-free and starts at 01.12.

That's it for the Lecture Series!
- We now have covered the basics of Python
- I hope you enjoyed the lecture and found it helpful
- If you have questions or feedback, please let me know!
- I wish you all the best for your studies and your career!

Literature

Interesting Books

- Downey, A. B. (2024). Think Python: How to think like a computer scientist (Third edition). O'Reilly. Link to free online version
- Elter, S. (2021). Schrödinger programmiert Python: Das etwas andere Fachbuch (1. Auflage). Rheinwerk Verlag.

. . .

For more interesting literature to learn more about Python, take a look at the literature list of this course.