Introduction to Vibe coding for non-programmers

Course code: VIBECD1

A practical course for anyone who wants to create their own application with the help of AI, even without previous programming experience. Participants will first understand the basic building blocks of the program and try out a simple algorithm created with AI. Then they will move on to agent development, in which they gradually create and extend their own application with multiple files and persistent data. The emphasis is on the result not being a one-time attempt, but a project that can be further developed: testing, Git as a safe history of changes, systematic debugging, working with context limits across sessions, project instructions in CLAUDE.md, documentation distribution and planning of major changes before implementation. Participants will not only take away a functional application, but above all practical habits and procedures that they can use in other and significantly larger projects. For the practical exercises, each participant must have their own account with access to Claude Code.

Who is the course intended for

The course is intended for anyone who wants to create their own applications with the help of AI - for beginners with no programming experience, managers, salespeople, specialists in operations, marketing, finance or HR, analysts, product owners, testers and developers who want to systematically involve AI coding agents in their work.
Practical exercises take place in Claude Code, but the principles learned can also be used in other agent development tools.

What we will teach you

- understand the basic building blocks of a program
- create a simple algorithm using AI
- verify the correctness of the code using tests and unit tests
- use Git as a secure project history
- create and gradually expand the application using an AI coding agent
- continue working on an existing project across sessions
- systematically find and fix errors
- work with the limits of the AI ??agent's context
- manage the work of the AI ??using project instructions and structured documentation
- plan major changes before their implementation
- prepare the project so that it can be followed by a new AI agent, a new session or another developer

Required entry knowledge

- Basic computer work
- Experience with programming or AI tools is not necessary
- Own account with access to Claude Code
- (a paid Claude Pro, Max, Team or Enterprise subscription, or a Claude Console account with API credit)

Course outline

Basic building blocks of the program

A simple algorithm created using AI

- Entering an AI chat task
- Transferring code to the program and running it
- Editing the result

Testing and unit tests

- Why test code created by AI
- Manual test and unit test
- Running tests

GOPAS Praha
Na Strži 2097/63
140 00 Praha 4 - Krč
Tel.: +420 226 201 390
info@gopas.cz

GOPAS Brno
Nové sady 996/25
602 00 Brno
Tel.: +420 530 513 590
info@gopas.cz

GOPAS Bratislava
Dr. Vladimíra Clementisa 10
Bratislava, 821 02
Tel.: +421 902 903 132
info@gopas.sk

Copyright © 2026 GOPAS, a.s.,
All rights reserved

Introduction to Vibe coding for non-programmers

Git as a safe project history

- Diff and commit
- Change history
- Reverting to the previous version

The first simple application using the AI ??coding agent

- Running Claude Code in a project
- Entering and creating an application
- Checking and approving changes
- General principle of the agent's work directly with project files

Working across sessions and continuing in an existing project

- Named sessions in Claude Code
- Returning to work in progress
- What the agent must find out again when continuing the project

Debugging

- Reproducing the error
- Finding the cause
- Repair
- Result verification

Multi-file applications with persistent data

- Splitting the application into multiple files
- Saving and loading data

Gradual expansion of the application using an AI coding agent

- Small changes instead of large interventions
- Continuous verification
- Commit after each functional change

Limits of context and work across sessions

- What is the AI ??agent context and why is it limited
- What can be lost during long work or a new session
- How to ensure that the agent knows important information about the project

Project instructions for an AI agent

- Why use persistent project instructions
- CLAUDE.md
- as their practical implementation in Claude Code
- Creating and maintaining project instructions
- Transferring the same principle to other AI coding agents

Project rules and how to work with a project

- Architectural rules
- Project conventions
- Commands for launching the application
- Commands and rules for testing

Structured project documentation

- Splitting information into
- requirements.md
- ,

GOPAS Praha
Na Strži 2097/63
140 00 Praha 4 - Krč
Tel.: +420 226 201 390
info@gopas.cz

GOPAS Brno
Nové sady 996/25
602 00 Brno
Tel.: +420 530 513 590
info@gopas.cz

GOPAS Bratislava
Dr. Vladimíra Clementisa 10
Bratislava, 821 02
Tel.: +421 902 903 132
info@gopas.sk

Copyright © 2026 GOPAS, a.s.,
All rights reserved

- architecture.md
- and
- testing.md
- Separating requirements, technical design and testing rules
- Referencing documentation from project instructions

Planning before implementing major changes

- Drafting a plan before changing the code
- Checking the proposed procedure
- Implementing according to the agreed plan

Handing over and continuing the project

- README.md
- and how to start the project
- Known limitations
- Handing over the project to a new session, another AI agent or another developer

GOPAS Praha
Na Strži 2097/63
140 00 Praha 4 - Krč
Tel.: +420 226 201 390
info@gopas.cz

GOPAS Brno
Nové sady 996/25
602 00 Brno
Tel.: +420 530 513 590
info@gopas.cz

GOPAS Bratislava
Dr. Vladimíra Clementisa 10
Bratislava, 821 02
Tel.: +421 902 903 132
info@gopas.sk

Copyright © 2026 GOPAS, a.s.,
All rights reserved