COMP 4991
ST: AI Tools for Software Development

Spring 2026
Instructor: Dr. Scott Fleming (sdflming@memphis.edu)
Lecture: Mon/Wed 2:20–3:45pm

Course Description

Students will learn how to use AI-based developer tools across the software development lifecycle, for example in coding, code reviewing, project management, automated testing, and security. The course will require significant software development practice, both with and without AI tools. Students will use their experiences to analyze the impact of AI tools on software productivity across individuals, teams, and organizations. Students will be expected to code in scripting languages (e.g., Python and JavaScript), set up development environments, read documentation, and learn new libraries and frameworks.

Prerequisite: COMP 3081 - CS3 Introduction to Software Design

Required Texts

No textbooks required. Students will read a selection of online sources that will be provided and include the following works.

- Andrew Begel. 2018. Creating Dev Specs. *Lecture Notes from INFO 461 Cooperative Software Development*. URL: https://ai-developer-tools.github.io/assets/pdfs/dev-spec-guidelines.pdf
- Huo Jingnan. 2025. Tech CEOs say the era of 'code by AI' is here. Some software engineers are skeptical. *NPR*. URL: https://www.npr.org/2025/10/21/nx-s1-5506141/ai-code-software-productivity-claims
- Chris Loy. 2025. The AI coding trap. URL: https://chrisloy.dev/post/2025/09/28/the-ai-coding-trap
- Veronica Pimenova, Sarah Fakhoury, Christian Bird, Margaret-Anne Storey, Madeline Endres. 2025. Good Vibrations? A Qualitative Study of Co-Creation, Communication, Flow, and Trust in Vibe Coding. DOI: 10.48550/arXiv.2509.12491 (Preprint)

- Peter Steinberger. 2025. Just Talk To It - the no-bs Way of Agentic Engineering. URL: https://steipete.me/posts/just-talk-to-it
- Various documents and tutorials from *Anthropic Academy*. URL: https://www.anthropic.com/learn
- Various documents and tutorial from *Cursor Learn*. URL https://cursor.com/en-US/learn

Required Laptop

Students will be required to bring a laptop computer to lecture and to have a computer to do the work required for the course.

Learning Goals

- Requirements Engineering
    - Learn how to do effective user discovery interviews to learn about the problems your users have, how they are willing to have you solve them, and what value they ascribe your solution
    - Learn how to get an LLM to create sound user stories that accurately reflect your planning goals
- Development Specification
    - Learn how to translate user stories into a structured development specification.
    - Learn how to guide an LLM to generate architecture diagrams, class diagrams, flow charts, and other technical artifacts.
    - Learn how to critically assess risks, interfaces, and technologies when planning software implementation.
- Frontend Development
    - Learn how to transform development specifications into a working user interface.
    - Learn how to use Figma's AI tools to create mockups.
    - Learn how to collaborate with LLMs to generate frontend code, while refining and debugging the output, with a mocked backend.
- Backend Development
    - Learn how to transform development specifications into a working backend.
    - Learn how to collaborate with LLMs to generate backend code, while refining and debugging the output, with a real or mocked frontend.

- Software Testing
    - Learn how to write effective automated software tests.
    - Learn how to collaborate with LLMs to generate unit, integration, and system tests.
- Software Deployment
    - Learn how to deploy web applications to the web.
    - Learn how to utilize LLMs to assist in the configuration, staging, and uploading of new versions of a web application to the web.

Course Structure

This class will consist of:

- **Mini-lectures.** Lectures will cover technical content and be delivered by the professors or guest lecturers.
- **In-class programming activities.** We will have pair programming activities and "mob" programming activities. These will be for learning techniques and tools that you will then use for your group project. Each student will lead at least one mob session.
- **Reflections.** For homework, you will write 500-word essays answering a set of assigned questions about your experience using AI so far.
- **Readings.** We may assign supplementary readings from online sources that are relevant to the activities or discussion.
- **In-class presentations and discussions.** Based on the reflections, groups will create short presentations, present them to the class, and lead a short discussion.
- **Group project.** During this course, you will build a web application using AI. The major milestones will be a vibe-coded MVP, requirements, backend implementation, frontend implementation, testing, and deployment.
- **Team meetings.** We will provide time teams to meet about their project. Participation is mandatory.

The course calendar follows a two-week cycle:

- **Monday:** Pair programming activity.
- **Wednesday:** Team meeting, mini-lecture, and mob programming activity.
- **Homework:** Reflection essay.
- **Monday:** Team meeting, mini-lecture, and mob programming activity.
- **Wednesday:** Presentations and discussions.
- **Project:** Two-week sprint milestone.

There will be no exams. The final exam day will be used for final project presentations.

Grading

Evaluation will be based on the following distribution:

- 46% in-class activities
- 42% course project
- 12% homework essays

The grading scale is as follows:

Letter Grade	Percent Score
A+	97–100
A	93–96
A-	90–92
B+	87–89
B	83–86
B-	80–82
C+	77–79
C	73–76
C-	70–72
D+	67–69
D	60–66
F	0–59

Attendance is required. Unexcused absences will result in a deduction in your overall course grade.

Schedule

Week	Topic
1 2	Vibe Coding
3 4	Requirements Engineering
5 6	Development Specification
7 8	Frontend Development
9 10	Backend Development
11 12	Software Testing

13 14	Software Deployment
15	Project Presentations