Documentation and Reflection

This semester, I used two different large language models (LLMs) — DeepSeek (AI 1) and ChatGPT (AI 2) — to help me complete several coding assignment like an expression evaluator written in Racket. I used both tools to identify bugs, fix syntax errors, and improve the structure of my program.

I used DeepSeek primarily for code generation and structure. It was helpful in generating functional code from scratch or based on problem descriptions. ChatGPT was mainly used for debugging and understanding errors, especially Racket syntax and logic mistakes.

DeepSeek was generally better at writing code quickly and suggesting complete functional blocks. However, it struggled with parentheses placement and often produced Racket code with subtle syntax errors. It was also not great at recognizing or explaining the cause of error messages in detail. ChatGPT, on the other hand, was excellent at reading Racket error messages, identifying missing else clauses, misplaced parentheses, and other subtle syntax issues. It provided not only the fix but also a clear explanation of what was wrong and why.

Using both LLMs helped me understand the importance of code structure and syntax, especially in a language like Racket where parentheses matter. They allowed me to iterate faster, learn from mistakes, and focus more on logic and design rather than being stuck on syntax. ChatGPT in particular improved my debugging skills by teaching me how to read error messages more effectively.

In some cases, relying too much on AI-generated code, especially from DeepSeek, led to time spent debugging avoidable errors. It sometimes gave me a false sense of correctness before I thoroughly tested the code. However, these experiences taught me to be more cautious and thorough with verification.

To use these tools more effectively, I need stronger debugging and testing skills, and a deeper understanding of the syntax rules of the programming language I'm using. Additionally, being able to ask clearer, more specific questions help get better responses from both AIs. Knowing when to verify and when to trust the AI is a critical skill I continue to develop. Furthermore, that won’t be enough if you don’t understand some of the programming concepts discussed in this class because it won’t allow you to ask the right questions and debugging code that you don’t understand could make things harder.

As the semester wraps up, the big picture I’m taking away from using AI tools like DeepSeek and ChatGPT is that they are incredibly powerful learning companions, not just coding assistants. When used wisely, they accelerate the debugging process, help you think through design problems, and offer different perspectives on solving technical challenges. But they are not replacements for understanding — they're supplements. They work best when paired with your own critical thinking.

Don't treat AI tools as crutches. Use them to learn why something works, not just how to make it work. Ask questions, verify outputs, and take time to understand the fixes they suggest. If something looks off, test it and dig deeper. Try using multiple AIs — I found that DeepSeek writes decent code quickly, but ChatGPT is much better at explaining problems and helping with debugging. Both have strengths, but neither is flawless.

For faculty designing courses, I suggest embracing these tools rather than avoiding them. Structure assignments in ways that encourage students to reflect on their use of LLMs — like this one — and build in checkpoints that require students to explain their code and decisions. That way, you promote learning and understanding, not just code submission. These tools aren't going away, and the classroom is the best place to teach students how to use them responsibly. Other classes even ask students to record a short video explaining the basics of their code and testing their code to ensure understanding. This would often require less testing, forcing students to memorize each term or concept by word, and instead encourage testing through projects and explanations.