Foundations Before Tools[1]*: What My Classroom Experiments Taught Me About AI*

[1] this text is prepared through human-AI collaboration based on the human provided content

Mustafa Mısır, Data and Computational Science, DKU, China
mustafa.misir@dukekunshan.edu.cn

Summary

This report documents my experience redesigning assessments across 5 **Computer Science** (**CS**) and **Data Science** (**DS**) courses in the presense of Generative AI (GenAI) tools. The courses include

- **STATS 302: Principles of Machine Learning**,
- **COMPSCI 308: Design and Analysis of Algorithms**,
- **COMPSCI 310: Introduction to Databases**,
- **COMPSCI 201: Introduction to Programming and Data Structures**, and
- a new Mini-Term course on **Co-Designing Algorithms with LLMs**.

The starting point was a clear observation. Students increasingly turn to AI for immediate solutions, often before attempting problems themselves. While this behavior is efficient to save time, it risks reducing opportunities for reflective thinking and meaningful understanding. The central challenge is not to ban AI as it is not completely possible, but to ensure students build strong **foundational knowledge** before using it as a tool, following appropriate assessment methods. Thus, the core principle guiding this redesign is to verify foundational knowledge before permitting tool use. Just as students must learn multiplication by hand before using a calculator, for instance, they must demonstrate understanding of programming logic before relying on AI code generation. The strategy is to create multiple evidence streams that AI cannot fabricate convincingly, such as live technical questioning, design justifications and peer teaching / learning. Then, transform AI from a shortcut into a validated collaborator.

Policy changes alone are insufficient to realize this goal. Telling students "do not use AI" relies only on instructions, which students may overlook under pressure. This is referred to a discursive change [Corbin et al., 2025]. What actually affects student behavior are structural changes. These reshape the assessment task so that genuine understanding is required for success.

With this focus, Table 1 summarizes these strategies and their application in my courses. The practices described here with the Six Assessment Redesign Pivotal Strategies (SARPS) framework [Chan and Colloton, 2024]. The table shows a shift from traditional methods to structural reforms. For example, under Integrating Multiple Assessment Methods, I enhanced existing weekly project meetings by changing the type of questions asked. Instead of asking 'what did you do,' I ask students to explain specific algorithm choices during code walkthroughs. Under Promote Authentic Assessments, I ask students to solve problems on paper before coding. Under Prioritising Feedback Over Grades, I plan to make certain weekly assignments optional and ungraded in the Machine Learning course to reduce panic-driven AI misuse. Under Embracing AI as a Learning Partner, the upcoming Mini-Term course structures the entire learning process around human-AI collaboration, teaching students to diagnose AI failures.

SARPS Strategy	**Core Idea**	**Course Application Example**
Integrate Multiple Assessment Methods	Diversify techniques to reduce over-reliance on one approach	Weekly project meetings with technical questioning in Machine Learning (STATS 302) and Databases (COMPSCI 310); Peer Learning Meetings with reflective reports in Programming (COMPSCI 201).
Promote Authentic Assessments	Shift to real-world problem solving with ambiguity or time pressure	Pen-paper problem solving before coding in Programming (COMPSCI 201)
Promote Academic Integrity and Genuineness	Ensure original thought remains intact despite AI assistance	Paper-based exams to verify foundational knowledge without AI access.
Embrace AI as a Learning Partner	Celebrate collaborative potential of human and machine	Mini-Term course on Co-Designing Algorithms with LLMs.
Prioritise Soft Skills/Human-Centric Competencies	Assess irreplaceable qualities like communication and thinking about learning	Reflection reports on peer learning discussions in Programming and Data Structures (COMPSCI 201); spontaneous technical questioning in project meetings in Machine Learning (STATS 302) and Databases (COMPSCI 310).
Prioritise Feedback Over Grades	Shift from grading to holistic feedback to reduce panic-driven AI misuse	Optional ungraded homework assignments with TA feedback in Machine Learning (STATS 302).

Table 1: Summary of Six Assessment Redesign Pivotal Strategies (SARPS).

Introduction

When I first looked at my course list for the upcoming year, I felt a stronger sense of responsibility than usual. I was keen to ensure students learned foundational knowledge strongly. However, I worried that AI could become an obstacle to this goal. I was preparing to teach 5 different classes. 4 of them, Introduction to Programming and Data Structures, Principles of Machine Learning, Design and Analysis of Algorithms and Introduction to Databases, are courses I have taught before, some multiple times[2]. I know their rhythms, their challenges and how students learn best in each. The last one is completely new though. It is an exploratory Mini-Term course titled *Co-Designing Algorithms with LLMs*[3], mainly targeting human-AI collaboration [Puerta-Beldarrain et al., 2025].

[2] https://mustafamisir.github.io/teach.html

[3] Large Language Models – https://www.ibm.com/think/topics/large-language-models

- STATS 302: Principles of Machine Learning[4]
- COMPSCI 308: Design and Analysis of Algorithms[5]
- COMPSCI 310: Introduction to Databases[6]
- COMPSCI 201: Introduction to Programming and Data Structures[7]
- MT-CoDA-LLM: Co-Designing Algorithms with LLMs[8]

My awareness of how deeply AI has changed learning became clearer last summer. I taught a 4-day intensive Introduction to Programming class to ~100 students, most with no prior programming experience. I noticed that almost every student turned to AI first for any programming question. Before trying a problem themselves, before asking a peer, before raising a hand, they asked AI. This was not necessarily laziness. It was a natural response to a tool that feels immediate and helpful. But it raised a critical question:

> *if students with no foundation rely on AI from the start, how do they learn to judge whether the answer makes sense?*

Research on Generative AI (GenAI) in STEM[9] education shows that students often use AI to copy and paste problems for direct solutions, with time efficiency being the primary motivator [Wang et al., 2025]. While this can feel productive, it risks 3 interconnected learning challenges: *cognitive offloading* [Gerlich, 2025], where mental work is delegated to external tools; *metacognitive laziness* [Fan et al., 2025], where reflective thinking about one's own reasoning is neglected; and a gradual reduction in *critical thinking* capacity [Lee et al., 2025]. My answer draws on two simple analogies. First, think of navigation apps. They are remarkably advanced and can guide you anywhere. But if you rely on them completely while driving or walking around, you may find yourself unable to navigate effectively when the app fails, loses signal or gives an incorrect turn [Dahmani and Bohbot, 2020]. You might even get lost on a road you have driven often because you never learned the route but mainly followed the instructions. Second, consider medical professionals. AI tools can give medical decisions faster [Di Fazio et al., 2026] and with impressive accuracy, some comparable to experts [Miranda et al., 2025, Takita et al., 2025]. And doctor-AI collaborations can offer further im-

[4] https://mustafamisir.github.io/stats302-f2025-s1.html

[5] https://mustafamisir.github.io/compsci308-f2025-s1.html

[6] https://mustafamisir.github.io/compsci310-f2025-s2.html

[7] https://mustafamisir.github.io/compsci201-s2026-s3.html

[8] https://mustafamisir.github.io/CoDA-LLM-s2026-MT.html

[9] Science, Technology, Engineering, and Mathematics

provements when used effectively[10]. Yet a doctor who delegates all judgment to AI risks allowing their own diagnostic skills to weaken over time [Monteith et al., 2026]. The expert human remains accountable for the final decision. AI in programming works the same way. It can generate code fast, but the learner remains responsible for the logic, the design, and the outcome. In that sense, *the goal is not to stop students from using AI. It is to help them become the experts who can judge, validate, and own the final result*.

[10] https://erictopol.substack.com/p/when-doctors-with-ai-are-outperformed

Building Strong Foundations

The biggest priority is ensuring students have strong foundational knowledge. If AI is used to write code, understanding the logic behind it matters. This matters not only because AI tools sometimes make mistakes, known as hallucinations [Jones, 2025], but also because relying on AI without foundational knowledge undermines the learning process [Favero et al., 2025]. To help avoid this, I adopted a principle similar to architecture and construction. Before building a structure, an architect draws detailed plans and confirms they are structurally sound. Only after the design is validated does actual construction begin. In the Introduction to Programming and Data Structures course, I encourage students to solve problems on paper first, often at the pseudocode (Figure 1) level, before writing any code.

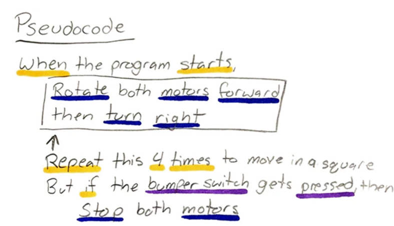

Figure 1: Pseudocode example – source: `https://education.vex.com/xyleme_content/loop-there-it-is-iq/web/be396427-5c31-4372-ad63-3bdd931ce3e9.html`

During lectures and lab sessions, I walk around and ask practical questions to confirm that students have thought through their approach before jumping into coding. I also suggest using simple text editors like Notepad during early learning stages, rather than full featured Integrated Development Environments (IDEs)[11] like VSCode[12]. Modern IDEs automatically fix syntax, warn about errors, suggest code completions and even implement a complete program from scratch. While powerful, these features can allow students to produce working code without truly understanding the logic. It is similar to a first-year primary school student asking a parent to complete homework assignments. The homework may receive a good grade, but the learning has not happened. Once foundational skills are secure, students are encouraged to transition to professional tools like VSCode, which improve efficiency and accuracy. When foundational understanding is proven, tools can be used with confidence, just like a radiologist who can interpret an X-ray as they know anatomy.

[11] https://aws.amazon.com/what-is/ide/

[12] https://code.visualstudio.com/

The Comfort and Risk of AI Interaction

I also noticed that AI tools make learning feel easier because they are always available, never judge you, and do not say "I am busy" or "That is a meaningless question." This makes it tempting to ask AI instead of a person. But there is a downside. Learning by yourself with AI means you miss out on real conversations. When you explain an idea to a classmate or instructor, you have to organize your thoughts, defend your reasoning, and hear questions that challenge your assumptions. AI does not necessarily do that. It gives answers, but it does not push you to think deeper, unless requested explicitly. To help students practice these essential skills, I encourage them to talk more, with peers, TAs, and me.

Multiple Ways to Show Knowledge

Once the foundation is set, multiple ways to show knowledge are needed. AI can write text, but it cannot fake a live conversation about why a specific design choice was made, unless it has been used actively during the conversation. In professional practice, for instance, developers or engineers must justify their decisions, not just deliver code. In the Introduction to Databases and Principles of Machine Learning courses, weekly project group meetings were held with a specific discussion methodology beyond simply accepting a project report at the end. During these meetings, technical questions are asked like "Explain this algorithm choice" rather than "What was done?" Code walkthrough happens where students point to the specific part of the code that implemented a requirement. This change reshapes the assessment. It ensures everyone is contributing and understands the work. It also brings us back into conversation, restoring the human connection that AI cannot replace. These meetings are for support, to help clarify thinking. This strategy matches the SARPS[13] framework category of Integrating Multiple Assessment Methods and the assessment type of Prioritising Human-Centric Competencies [Chan and Colloton, 2024]. It focuses on communication and awareness of one's own thinking.

[13] Six Assessment Redesign Pivotal Strategies

In the Introduction to Programming and Data Structures course, I have been also using this approach, yet for homework assignments and without any meetings. Students submit design explanations and failure reasoning, as shown in Figure 2. For instance, they must explain why and when their code may fail, not just show the fixed version. The TA provides weekly feedback. If a submission is unclear or needs review, there is a discussion to understand the student's thinking. This adds another layer of verification.

- **Design explanation.** Briefly explain your overall approach. Describe the data structures (data containers) such as arrays and the algorithms (step-by-step procedures) you used. Explain why you chose this approach over other possible alternatives and why your choice is appropriate for this problem. Include reasons why this approach is simpler, more efficient, or more reliable etc. than other approaches. Example:

```
/* Design explanation:
   1. I used an array to store input numbers because arrays allow
      easy access by position and can handle any input size.
   2. I calculate the sum first and then divide by the number of
      elements to get the average.
   3. I chose this approach over using separate variables for each value
      because it works for any number of inputs, keeps the code simple,
      and clearly demonstrates the algorithm for calculating averages. */
```

- **Failure reasoning or guarantees.** Identify at least one scenario, if exists, where your code may fail or produce incorrect results and explain why. If your code works for all valid inputs, explicitly state that and explain why it cannot fail. Example:

```
/* This code fails if the array is empty because dividing by zero.
   For all non-empty arrays it always produces a result because
   sum and array length are correctly calculated */
```

Figure 2: A partial screenshot from a homework assignment document stating design explanation and failure reasoning / guarantees requirements.

Structured Peer Learning Meetings

To further balance the tendency to work alone with AI, I introduced structured Peer Learning Meetings in the Introduction to Programming and Data Structures course. Students work in pre-determined pairs, formed either by mutual choice or assigned according to course enrollment. Each pair meets weekly for ∼30-60 minutes to review recent course materials, examples and conceptual difficulties. After each meeting, each student submits an individual reflective report identifying specific concepts that became clearer through discussion, at least one misconception corrected during the meeting and how their understanding changed compared to before the discussion. Reports must be written in the student's own words and focus on learning outcomes rather than meeting summaries. This activity is collaborative but non-graded as a group, reducing performance pressure while encouraging genuine engagement. The idea relies on my personal experience when I was a student that teaching a concept to a peer strengthens one's own understanding far more than passive review. It also restores the human element that AI cannot replicate. This includes the discomfort of not knowing, the satisfaction of mutual discovery, and the accountability of explaining one's reasoning to another person.

AI as a Learning Partner

I am also planning to make AI a learning partner rather than a forbidden or limited-use tool. In the upcoming Mini-Term course on Co-Designing Algorithms with LLMs, the entire learning process is structured around human-AI collaboration. This course is a new initiative, so outcomes cannot yet be reported. But the design reflects a belief developed through teaching other courses. Teaching learners to work with AI is more valuable than teaching them to work without it. Over 4 days, the process moves from algorithm intuition to prompt[14] refinement, and then into generate-test-revise loops. Students learn to diagnose failures when the AI gives wrong answers. This is designed to teach validation skills. Learners see that AI hallucinates, and that human judgment is the final filter. By integrating this approach into an Algorithms[15] course, the aim is to show that AI is powerful, but it requires a skilled human to guide it. This aligns with the SARPS strategy of Embracing AI as a Learning Partner, turning the tool into a collaborator rather than a shortcut [Chan and Colloton, 2024].

[14] https://www.ibm.com/think/topics/prompt-engineering

[15] the future idea is on how to incorporate this into COMPSCI 308: Design and Analysis of Algorithms

Shifting Focus from Grades to Feedback

Another significant change involves shifting the focus from grades to feedback. Homework assignments are meant for practice. They help internalize concepts beyond reading a textbook or listening to a lecture. But students often feel overwhelmed by multiple deadlines. When overwhelmed, shortcuts look tempting. AI offers a fast path to a submission. At the same time, verifying whether a homework solution is heavily AI-generated is extremely difficult without holding impractical one-on-one meetings for every student. The solution, planned for the Principles of Machine Learning course if to be taught next time, is to make certain weekly assignments optional and ungraded. Students who submit will still receive detailed TA feedback to support their learning. Students who do not submit will not receive personalized feedback, but they will have access to solution keys for self-study. The logic is that when the pressure to earn a grade is removed, the incentive to misuse AI diminishes. Students who genuinely want to practice are expected to engage. Students who are struggling can focus on core concepts without panic. This preserves the learning purpose of homework assignments while reducing the temptation to outsource thinking to AI. This reflects the SARPS strategy of Prioritising Feedback Over Grades [Chan and Colloton, 2024].

Transparency and Clear Boundaries

Finally, transparency about what works and what does not is important. All my course syllabi include certain AI use procedures. Specifications exist for where AI is permitted and where it is not. Yet my personal observations show that students still find ways to use AI across many tasks. It is essentially an observation about how policies alone function. When assessment relies only on instructions, it creates a discursive change. This means modifying only the rules or policies. Students remain free to interpret, adapt, or overlook those instructions when pressure mounts. What actually shifts behavior are structural changes. These are modifications that reshape the mechanics of the task [Corbin et al., 2025]. Table 2 shows the difference between these two approaches.

Traditional Practices		**Structural Reforms**
Rules & Restrictions		Verification & Collaboration
Detection & Policing		Evidence Over Detection
AI as Threat		AI as Partner
Discursive Changes: instructions only, easily ignored	$\longrightarrow$	*Structural Changes:* task mechanics reshaped, requires understanding

Table 2: Assessment redesign in the GenAI era: contrasting traditional practices with structural reforms. The key distinction is that discursive changes modify only instructions while structural changes reshape task mechanics.

For example, in the Introduction to Databases course, clarification exists that AI can help with User Interface (UI) design (Figure 3) or generating synthetic data. However, explicitly course related schema design decisions and normalization tradeoffs remain human-responsible judgments. This clarity does not physically prevent AI use in protected areas. But it does something equally important. It shows where to focus effort. Tasks outside core learning objectives become opportunities to use AI for efficiency. Tasks inside core objectives remain spaces for own reasoning and growth. When boundaries are clear, AI can be used creatively in safe zones while building own expertise where it matters most. This approach relates to the **AI Assessment Scale** [Furze et al., 2024, Perkins et al., 2024] which define levels of AI permission from prohibited to fully integrated. And when AI is used, ownership of the final judgment is maintained, just as a doctor signs off on a diagnosis after reviewing all the evidence.

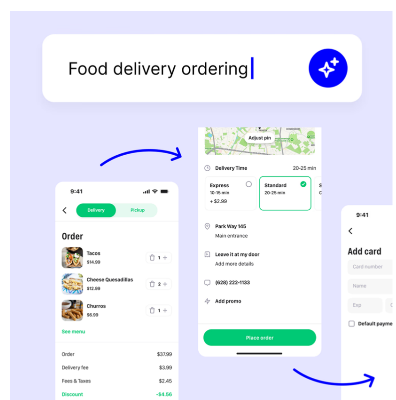

Figure 3: An example for AI-driven automated UI generation by simply explaining what you want – `https://www.banani.co/blog/ai-for-ui-design-and-wireframes`

References

CKY Chan and T Colloton. Redesigning assessment in the ai era. *Generative AI in higher education: The ChatGPT effect*, pages 87–126, 2024.

Thomas Corbin, Phillip Dawson, and Danny Liu. Talk is cheap: Why structural assessment changes are needed for a time of genai. *Assessment & Evaluation in Higher Education*, 50(7):1087–1097, 2025.

Louisa Dahmani and Véronique D Bohbot. Habitual use of gps negatively impacts spatial memory during self-guided navigation. *Scientific reports*, 10(1):6310, 2020.

Nicola Di Fazio, Christian Zanza, Yaroslava Longhitano, Antonio Voza, Roberto Balagna, Sabino Mosca, Pietro Balagna, Riccardo Rossignoli, Sara Cerenzia, Giuseppe Bertozzi, et al. Artificial intelligence for early diagnosis in emergency department. *Journal of Anesthesia, Analgesia and Critical Care*, 6(1):7, 2026.

Yizhou Fan, Luzhen Tang, Huixiao Le, Kejie Shen, Shufang Tan, Yueying Zhao, Yuan Shen, Xinyu Li, and Dragan Gašević. Beware of metacognitive laziness: Effects of generative artificial intelligence on learning motivation, processes, and performance. *British Journal of Educational Technology*, 56(2):489–530, 2025.

Lucile Favero, Juan-Antonio Pérez-Ortiz, Tanja Käser, and Nuria Oliver. Do ai tutors empower or enslave learners? toward a critical use of ai in education. *arXiv preprint arXiv:2507.06878*, 2025.

Leon Furze, Mike Perkins, Jasper Roe, and Jason MacVaugh. The ai assessment scale (aias) in action: A pilot implementation of genai-supported assessment. *Australasian Journal of Educational Technology*, 40(4):38–55, 2024.

Michael Gerlich. Ai tools in society: Impacts on cognitive offloading and the future of critical thinking. *Societies*, 15(1):6, 2025.

Nicola Jones. Ai hallucinations can't be stopped—but these techniques can limit their damage. *Nature*, 637(8047):778–780, 2025.

Hao-Ping Lee, Advait Sarkar, Lev Tankelevitch, Ian Drosos, Sean Rintel, Richard Banks, and Nicholas Wilson. The impact of generative ai on critical thinking: Self-reported reductions in cognitive effort and confidence effects from a survey of knowledge workers. In *Proceedings of the 2025 CHI conference on human factors in computing systems*, pages 1–22, 2025.

Joana Miranda, Raquel Pereira-Silva, João Guichard, Jorge Meneses, Andreia Neves Carreira, and Daniela Seixas. Artificial intelligence outperforms physicians in general medical knowledge, except in the paediatrics domain: A cross-sectional study. *Bioengineering*, 12 (6):653, 2025.

Scott Monteith, Tasha Glenn, John Richard Geddes, Peter C Whybrow, Eric D Achtyes, Rita Bauer, and Michael Bauer. Artificial intelligence and deskilling in medicine. *The British Journal of Psychiatry*, pages 1–3, 2026.

Mike Perkins, Leon Furze, Jasper Roe, and Jason MacVaugh. The artificial intelligence assessment scale (aias): A framework for ethical integration of generative ai in educational assessment. *Journal of University Teaching and Learning Practice*, 21(6):49–66, 2024.

Maite Puerta-Beldarrain, Oihane Gómez-Carmona, Rubén Sánchez-Corcuera, Diego Casado-Mansilla, Diego Lopez-de Ipina, and Liming Chen. A multifaceted vision of the human-ai collaboration: A comprehensive review. *IEEE Access*, 13:29375–29405, 2025.

Hirotaka Takita, Daijiro Kabata, Shannon L Walston, Hiroyuki Tatekawa, Kenichi Saito, Yasushi Tsujimoto, Yukio Miki, and Daiju Ueda. A systematic review and meta-analysis of diagnostic performance comparison between generative ai and physicians. *npj Digital Medicine*, 8(1):175, 2025.

Karen D Wang, Zhangyang Wu, L'Nard Tufts, Carl Wieman, Shima Salehi, and Nick Haber. Scaffold or crutch? examining college students' use and views of generative ai tools for stem education. In *2025 IEEE Global Engineering Education Conference (EDUCON)*, pages 1–10. IEEE, 2025.