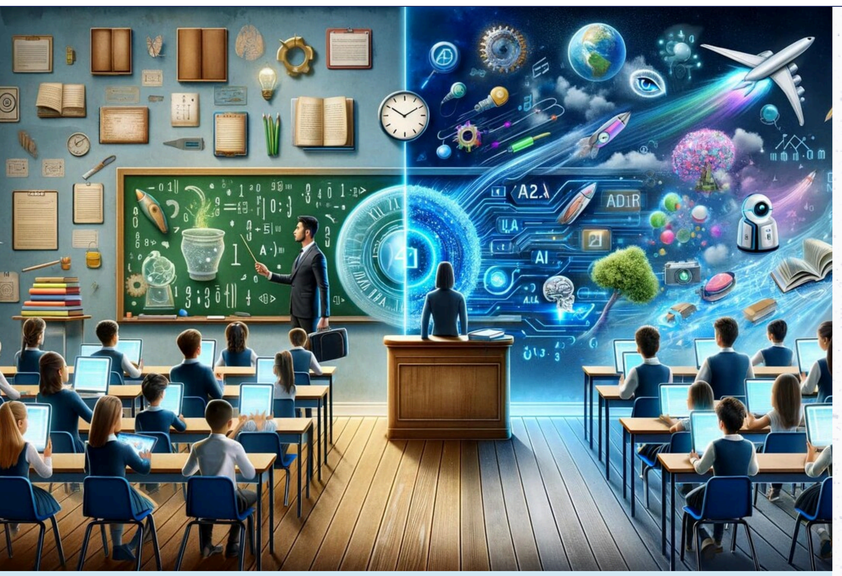
DR. SUKHCHANDAN

Assistant Teaching Professor
Department of Computing Science
Thompson Rivers University

Teaching, Like Code

From Lecture Design to Post-Class Refactoring: Instructor Uses of AI in Computing Science

How instructors think with AI
Before and After class



01. Introduction

In computing science education, instructors routinely apply practices such as debugging, testing, and refactoring—yet lectures are often treated as static performances rather than evolving instructional artifacts. While current discussions of AI in education emphasize student use, this poster shifts focus to instructor practice, exploring how AI can support lesson planning before lectures and reflection after lectures without redesigning courses or automating pedagogy.

02. Objective

To identify concrete, instructor-driven uses of AI that support:

- lecture clarity
- pacing and conceptual alignment
- reflective improvement across course offerings

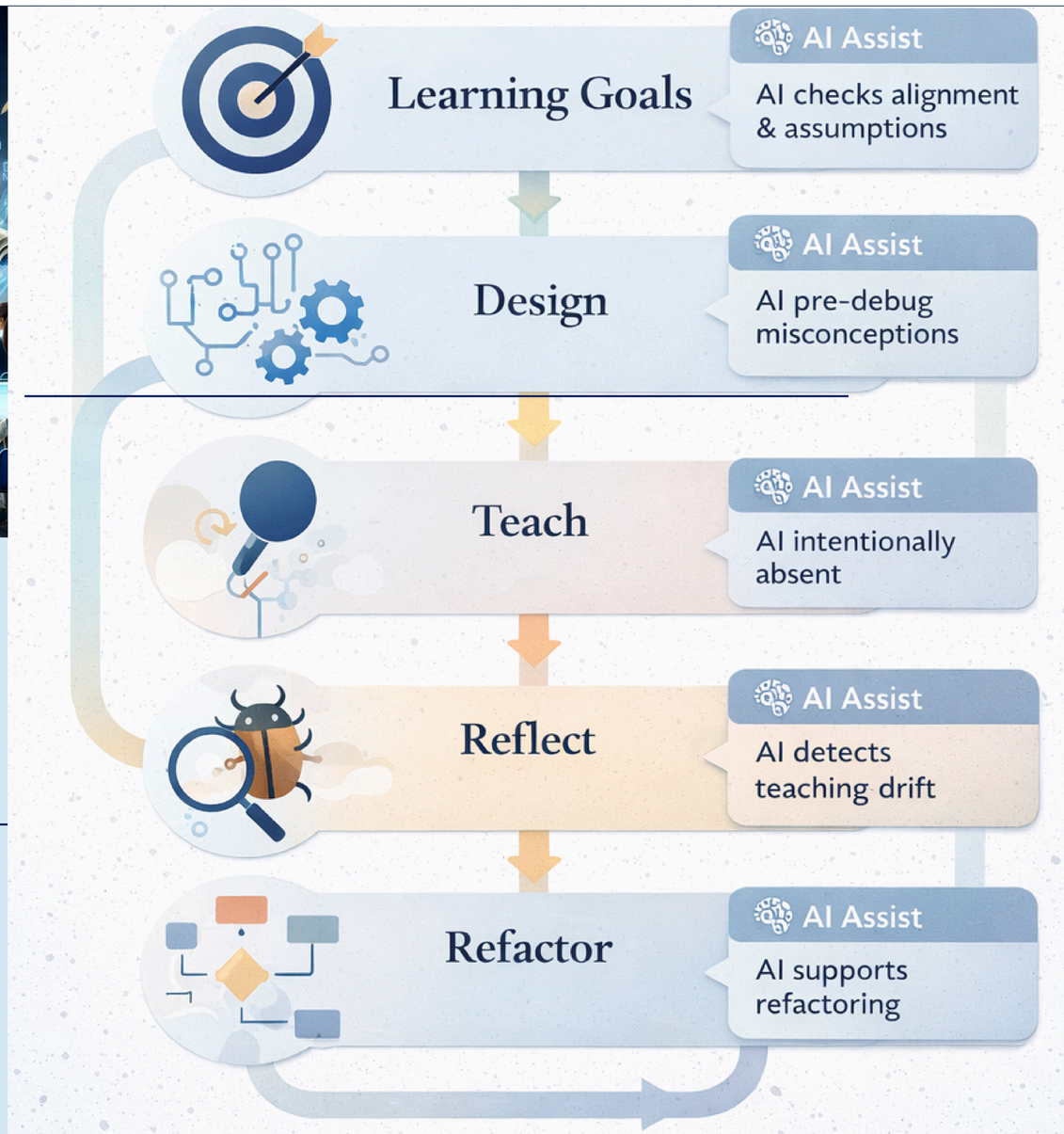
03. Methodology

This work is based on reflective teaching practice in undergraduate computing science courses. Instructor observations were gathered through iterative lesson planning, in-class delivery, and post-lecture reflection across multiple offerings.

Approach includes:

- Instructor-led lesson design
- Structured reflection after lectures
- Iterative refinement of explanations and examples

Instructor Teaching Pipeline



04. Instructor Practices Using AI

AI was intentionally used at two leverage points:

Before the Lecture

- Pre-debugging common misconceptions
- Generating counter-examples and edge cases
- Simulating pacing for novice learners
- Comparing multiple mental models
- Anticipating strong student questions

After the Lecture

- Detecting a mismatch between intent and delivery
- Refactoring explanations that failed
- Designing micro-clarifications
- Surfacing hidden assumptions

Important: AI-supported instructor thinking — it did not replace pedagogical judgment.

05. Conclusion

Treating teaching as an iterative system—much like code—allows instructors to leverage AI as a planning and reflection tool, not an automation engine. Small, concrete uses of AI before and after lectures can meaningfully improve teaching practice while preserving instructor autonomy and disciplinary integrity.