

DALLIN DAHL

dallinjdahl@gmail.com | github.com/dallinjdahl

TEACHING STATEMENT

I believe that understanding comes primarily in one of two ways: through hands-on experience, and by analogy to experience. Condensing many analogies results in a broadly applicable abstraction. The goal of education is to communicate those abstractions so students can then draw analogies. Abstractions are best taught by analogy to personal experiences, previous or current, and those abstractions enable understanding and application across diverse domains.

The basic nature of tooling, from the Japanese hand-plane to synchronous circuits to compilers, is all about taking first-hand experience, abstracting it, and applying it to a broader domain. The hand plane codifies expert experience about angle, pressure, and grain into a tool that even a novice can start to use. Synchronous circuits and compilers do the same for digital design and programming. Understanding is the primordial example of this pattern, applying and compressing our tacit knowledge.

I think that the constructionist and classical approaches fit this model well. Constructionism gives the students the opportunity to increase the portfolio of experiences from which they can draw analogies and build abstractions. Studying foundational works gives students experiences specifically in abstraction and analogy. Together they form a corpus of experiences and analogies that is robust to changes in technologies and paradigms.

Concretely, this means that courses I structure are generally built around group projects, in-class discussions of those projects, and foundational texts related to the topic. I work hard to listen to questions and understand the underlying conceptual framework that motivated those questions, and attempt to correct core conceptual issues whenever I can. The questions are the primary indicator of Vygotsky's zone of proximal development, and illuminate which experiences or analogies would best benefit the student.

Another indicator of the student's understanding is the assessment. When used, assessments should be reflective of the learning process, and not the products learning enables. AI specifically makes Goodhart's law increasingly relevant as formulated by Marilyn Strathern: "when a measure becomes a target, it ceases to be a good measure." This gives us the opportunity to clarify and distill our targets. Rather than try to tightly control the environment in which we assess, it is more effective to pick assessments that are immune to interference from AI and other sources. Activities such as in-class presentations, discussions, and critiques are rich displays of understanding. Other highly informative assessments include naming instances of a pattern, revising and debugging on the spot, and negotiating boundaries of a large project over time. These could also provide artifacts that motivate presentations and discussions.

On one occasion, I was helping a student struggling with their approach to some mathematics work. I noticed that the student was trying to intuit the answer without manipulating the problem into a clearer form. Inspired by the game-

theoretic semantics of proof search, I leveraged the student's prior exposure to games. The key features of games are the current state, and the set of available moves. Thinking of the problem in that framework allowed the student to focus on the important immediate decisions, without resorting to guessing and checking the final answer. The student later reported that they were able to address much more complicated problems in an assessment leveraging the same analogy.

On another occasion, I was assisting a student struggling with their mental model of C, specifically regarding pointers. They had some confusion about whether they needed to dereference the value, and how many times. They were thinking about the program as a puzzle: all they needed to do was keep turning the pieces around until they fit. By making an analogy to houses and their addresses, I was able to help the student reason through the building of the program logically.

When I teach, success happens when I listen. If I hear a student explaining why linear logic enables compile-time memory safety, I am confident they can apply formal reasoning to engineering concerns. When I hear a student ask a question about how to guarantee an optimization preserves the semantics of a circuit or a program, I know they will carefully consider the impact of a refactor. Listening allows me to tailor experiences and analogies to help students gain ownership over abstract concepts.