AGILE TEACHING

Teaching principles that correspond to agile development principles:

Teaching principle	Agile principle
Students need to see progress and value in their learning, promptly and often. Engage students in work that makes ongoing use of new learning, frequently illustrating its relevance and utility. Assess frequently.	Satisfy the customer through early and continuous delivery of valuable software. Deliver working software frequently, from a couple of weeks to a couple of months, with a preference to the shorter timescale.
Meet the student where they are, and be willing to deviate from your plan in response to student needs.	Welcome changing requirements, even late in development, and harness change for the customer's competitive advantage.
Maintain daily two-way communication with students and frequent communication with parents.	Business people and developers must work together daily throughout the project.
Nurture intrinsic motivation. Be curious about students' existing interests. Identify support and scaffolding that students need, and provide it. Trust that students want to learn and to be helpful and competent.	Build projects around motivated individuals. Give them the environment and support they need, and trust them to get the job done.
The best opportunity to observe, assess, and respond to students' needs is in face-to-face interactions during class.	The most efficient and effective method of conveying information to and within a development team is face-to-face conversation.
Usable, powerful, and broadly applicable knowledge, and transferable skills, are the primary measures of progress.	Working software is the primary measure of progress.
Teaching should promote metacognition and self-sustaining learning behaviors. The teacher must model sustainable behaviors — not compensating for student behaviors by adopting unsustainable and unhealthy responses — and the students must be learning sustainable behaviors.	Agile processes promote sustainable development. The sponsors, developers, and users should be able to maintain a constant pace indefinitely.
Rigor and reinforcement matter; attention to good practices (in work and learning) produces compounding benefits.	Continuous attention to technical excellence and good design enhances agility.
Simplicity—the art of eliminating distractions and spurious associations—is essential.	Simplicity—the art of maximizing the amount of work not done—is essential.
The most fruitful learning comes from listening to students' actual discussion and questions.	The best architectures, requirements, and designs emerge from self-organizing teams.
Students write weekly reflective journal entries or check-in emails; the teacher reflects on this and other student output, and adjusts teaching behavior and classroom activity accordingly.	At regular intervals, the team reflects on how to become more effective, then tunes and adjusts its behavior accordingly.