

# Larry D. Pyeatt

## Statement of Teaching Goals

---

I take teaching very seriously and strive to do the best job that I can. I have worked in industry and routinely bring that experience to the classroom to help prepare students to enter the work-force. Several students have attributed their success in industry to taking one or more of my classes. My overall philosophy of teaching can be described as follows:

**Continuous Improvement:** Not only is Computer Science a rapidly changing field, but new pedagogies are being developed all of the time. If a course does not change, then it becomes outdated. This is true of any field, but especially true for Computer Science. Thus, I work for continuous improvement in my course materials, content, and teaching style. This applies to not only the classes that I am teaching, but also to my duty to help determine and improve the curriculum for the department.

**Active Learning:** Active learning gets the student involved so that they learn the material at a deeper level than rote memorisation. In-class discussions and course projects are excellent ways to involve the students. The homework and projects should be chosen carefully to reinforce the most important concepts in the course. As students progress and mature, they should take more of the responsibility for learning. At some point, they can become their own teachers. That is the point at which they are truly educated.

**Appropriate Rewards:** Students should get the grade they earn. What many students want is to get the highest possible grades for the least amount of work. That is natural and should be expected. However, teachers have a responsibility to display fairness and integrity. It is important to set expectations, tell the students what the expectations are, and tie grades to how well the students meet those expectations.

Courses that I enjoy teaching include robotics, artificial intelligence, digital logic, architecture, operating systems, assembly language, and real-time systems. Courses that I would like to teach if given the opportunity include genetic algorithms, speech recognition, planning, machine vision, Markov decision processes, discrete mathematics, data structures, system administration, and compiler construction. In addition to these preferences, I am competent and willing to teach any traditional computer science course at either the undergraduate or graduate level.

My greatest teaching achievement involves a student who had a low GPA and was in danger of dropping out of the program. He confided in me that his dream was to be a mission controller for NASA. He also indicated that he was interested in robotics, so I told him that I would work with him on two conditions: he was to make a 4.0 GPA in the coming semester, and meet with me weekly for an independent study in reinforcement learning. At the end of the semester, he had achieved all A's and had a good basic understanding of reinforcement learning. More than that, our relationship had developed into a mentorship. By Fall of his senior year, he was doing research. He published his first paper as a senior and published another in his first year of graduate school. Not only did he blossom academically, but he also decided to work towards a PhD.

About one year into his dissertation, we were working on a research project with a group from NASA and he got the opportunity of a lifetime. He was offered a position as a mission controller on the International Space Station. I was sad to see him take the position, but also happy for him. Not many people get to achieve their dreams. My mentorship of him has given me a new perspective on teaching and advising: Some students need a teacher to get them interested and involved, and I can be that teacher. Nothing could be more personally rewarding.