Teaching Philosophy

I am of the opinion that teaching pedagogy can be separated into two main categories: technical-based mentorship and morale-based mentorship. Although these categories may seem separate, they are not mutually exclusive. Technical-based mentorship focuses on the "meat and potatoes" of what mentorship is. This is usually what one sees in a mentor: someone who helps students learn the practical and scientific skills of chemistry. Technical-based mentorship is spotlighted in classroom settings, where students gain direct knowledge and application for a given subject. This mentorship is also measurable through student testing and assignment grades. Morale-based mentorship, on the other hand, is a far more abstract beast to tackle. It cannot be measured directly, instead observed through teacher-student conversations where the teacher must determine (both objectively and subjectively) how the student is doing. Although there is a heavy focus on technical-based mentorship in student success, this success cannot be achieved without support, motivation, and patience. It is far easier to adequately accommodate a student with a broken arm, for example, by providing them a scribe for lectures, than to adequately accommodate a student who is struggling with burnout, academic-related anxieties, or personal struggles. Although professors may not be able to help a student with personal struggles, we may still act as morale supports to ensure that what is within our control may benefit a struggling student. It is at this intersection of technical and morale-based teaching where we as teachers and mentors can find the most success for our students.

Goals

My goal as a lecturer is simple: help students succeed. Although "simple", this success can take many forms. Finding how a student defines their success can be as straightforward as presenting students with a survey in the first class that asks "flash forward to the end of the semester. How would you describe a successful semester?" Sometimes it is as simple as helping a highly motivated student get an A in the course. Sometimes success is just encouraging students to attend lectures. It is clear the first goal relies heavily on technical-based mentorship; a student who understands the science and lecture material should be able to have their grades reflect this. The second example, however, falls into morale-based mentorship. There are ways to get students to come to class without morale-based mentorship, by using weekly lecture quizzes, for example. However, what may be more effective is creating a welcoming, safe, and interesting classroom environment. This last component can be achieved through various means, such as creating interactive lectures that get students excited about the lecture material, designing lectures and projects around real-life scenarios, and introducing general questions throughout lectures to keep students curious. My goal is to have students reach their own version of success, both inside and out of the classroom. For some students, this may be getting a 4.0 GPA for the semester. For others, it may simply be passing or showing up to class despite external challenges. No matter what this success looks like, it should be celebrated by both the student and the teacher.

Computational chemistry is often inaccessible to undergrads who do not have prior programming experience. I believe that developing a class to discuss the basics of programming for non-computer science majors can be incredibly helpful. During my own undergraduate experience, I had taken a Python for Mathematics class, aimed at students with little to no programming experience. This class was vital in helping me understand the basic properties of the Python programming language, as well as programmatic problem solving. Introducing a class like this

(Python for STEM majors, for example) can help students develop computational skills they would not otherwise have. This class would be offered to upper-level undergraduates and would include both general programming lectures as well as lectures where basic coding skills are applied to STEM-specific problems regarding chemistry and physics. By introducing students to Python, a free programming language with extensive resources, students will gain skills that are essential for the current technology age.

For highly motivated students who want to get involved in computational research, I highly recommend the development of a computational chemistry boot camp. This boot camp will be semi-guided and will contain modules on basic programming skills (a condensed version of the class described above), the Linux operating system (which many high-performance computing centers use), and chemical software such as Quantum Espresso, CP2K, or VASP. By the end of this boot camp, students should be able to run basic calculations and analyze their data using code that they have written themselves. This will not only help students learn the ropes of computational chemistry but will provide them with an opportunity to see if they like the research area. During this boot camp and continuing throughout the semester, group meetings can be structured so that every other week, for example, a higher-level computational topic is discussed in lecture form such as density functional theory or self-consistent field calculations. By structuring group meetings in this way, it will help students grasp concepts that are often not discussed in an undergraduate quantum chemistry course, while creating a space for students to ask both conceptual and technical questions and help to marry the two.

Ideas about pedagogical approaches

As each student learns in a unique way, pedagogical approaches should also be tailored, as much as is feasible, to each student. There are five key pillars of my pedagogy: 1) Provide various perspectives on problems; 2) Remove any singling out of students in the classroom; 3) Introduce problems and solutions as a team effort between students and faculty; 4) a knowledge-focused approach to solving problems; and 5) encouraging learning from mistakes.

1. Provide various perspectives on problems.

The first pillar is an example of technical-based teaching. Oftentimes, a student may struggle if the only resources they have for a class are the lecture slides and corresponding textbook. By opening up students to other perspectives, they may find something that works for them better while still reaching technical benchmarks. Although ChatGPT has become a common tool for students, it may not always provide accurate information. By steering students towards better tools, such as chemistry libre texts, YouTube videos like the organic chemistry tutor, and online interactive periodic tables, students can gain other perspectives while ensuring the information they absorb is accurate.

2. Avoiding singling out students.

The second pillar, avoiding singling out students unless necessary, has been shown to help increase class participation, even in larger class sizes. This can be accomplished by asking questions in a lecture to the entire class instead of just to specific students who choose to participate. I have had much success in implementing “easy” questions into lectures, where students are encouraged to shout out answers. Over time, I have seen more and more students participate in these types of interactions. An example of this is to ask students what the atomic number of an element is with the periodic table in the background. Another example is to ask a class yes/no either/or questions (e.g. in a quantum mechanics class, “if we know the position of an electron, do we know its momentum?”). Not only have I used this method to increase class participation, but it increases student confidence as well.

It is important to note, however, that this is not to discourage one-on-one mentorship with students. On the contrary, in using these techniques I have seen an increase in students who stay after class to discuss topics with me, along with increased turnout during office hours. It is important during these smaller-group environments to both make students feel like they are getting personalized attention while generalizing questions and challenges to a broader group. In using this method, students are indirectly encouraged to speak with me one-on-one, without any pressures to do so.

3. *Introduce problems and solutions as a team effort.*

The third pillar, introducing problems and solutions as a team effort (morale-based mentorship), has been by far the most effective pedagogy I have personally used. By removing the word “you” from conversations regarding challenges, it helps to lower any anxieties about seeking help. Instead, I simply rephrase, focusing not on the student but the challenge at hand and show that we will work through it together. Many years ago, I had watched a TED talk that described how language shapes how we think. The example used in the video was about a man who knocks over a vase: in English, we say just that: “*he* knocked over the vase” but in other languages, it can be phrased like “the vase got knocked over”. Here, we see that the man who knocked over the vase is removed from the action. We can apply this same logic to teaching. Instead of saying “you did this problem wrong” we can say “it doesn’t look like this was correct, let’s figure out a way to work through it together”. This simple rephrasing removes the student from the challenge at hand and reframes it into a group effort to resolve the issue. I had taught an undergraduate general chemistry lab where, when asked a question, I would begin with “well, let’s think about this…”. Before the halfway point in the semester, my students began copying me. They would ask a question, and before I had time to respond they would parrot “well, let’s think about this”. This simple phrase helped students feel comfortable asking questions and grew their confidence in answering them. Another example of this is the introduction of “what do *we* know?”. This immediately introduces a challenge as a collaborative effort, and, like the second pillar, it provides a foundation of “easy” topics a student knows. Although “we” is used, the conversation is led by the student with me asking guided questions to reach a final conclusion.

4. *Knowledge-focused approach.*

The fourth pillar, a knowledge-focused approach to challenges, is both a technical and morale-based form of mentorship. By stating the “obvious” about a problem, it builds a student’s confidence. Many students can get overwhelmed by demanding course loads and class requirements. Many students may say “I don’t know” when asked a question. Take, for example, “what is the molecular geometry of water?”. If a student replied that they didn’t know, the first step would be to ask what water is made of (hydrogen and oxygen). After this knowledge is established, one can increase the complexity by asking how many valence electrons hydrogen and oxygen have (one and six, respectively). We can then keep increasing the knowledge complexity until an answer to the original question can be formed: Will oxygen form a single or double bond with hydrogen (single; also see pillar two for either/or questions), how many electrons are in a single bond (two), how many lone pairs would oxygen have, and so on and so forth.

Knowledge, ultimately, cannot be built without a strong foundation. Ensuring a strong foundation is key to student success from a technical perspective. By focusing on what a student knows instead of immediately addressing what the student doesn't know, teachers and mentors can help build confidence, avoid anxious reactions to material, and encourage student-focused problem solving. I also heavily encourage my students to use what I call the Know-to-Need system when solving problems. This method consists of two parts: First, when faced with a problem (whether it be a test, quiz, or homework problem) a student should write down what they know. Often, this involves jotting down a list of information presented in the question itself, including any extra relevant information provided by supplementary material such as a periodic table, a formula, or constants. Just as previously stated, this helps a student tackle a problem first with confidence. Then, the question needed to be answered should be written (e.g., "weight of carbon dioxide = ?"). This is the "Need" aspect of Know-to-Need. By doing this, it turns a lengthy word problem into an easily digestible task to tackle. Whether it be multiple choice or open-ended questions, this method has shown success.

5. *Encourage learning from mistakes.*

The fifth pillar may, at the surface, seem to contradict pillar four. However, both can exist harmoniously. Learning from mistakes involves confronting confirmation bias in the classroom. This can be achieved through guided exercises where students are given a sample problem and incorrect solution. Then, students can be asked what the example does wrong (maybe, for example, units are incorrect, or a chemical reaction was not balanced correctly). These questions can be posed with yes/no questions as described in the second pillar to encourage group participation and to help guide students towards correct answers and ways of thinking. By using the Know-to-Need strategy outlined in the fourth pillar, it should help students recognize where a problem goes awry. By introducing this method, students should be able to address common mistakes and learn from them, whereas in a traditional classroom environment, students are expected to figure out mistakes on their own after receiving graded work if they do not take the initiative to discuss errors during office hours or over email.