There was a recent article published in *Nature* titled, "A mobile robotic chemist," that has been receiving attention in mainstream media. Burger et al. had developed a robot with the intention to "automate the researcher."[1] The robot could perform 688 experiments over a ten day period, which Professor Andrew Cooper explains could "lose their whole PhD".[2] While there are of course shortcomings to the current application of robots to do experiments, it does make one consider how far away a future is where even research can be automated and done by machines.

For a while now, we've acknowledged that advancing machinery and technology were coming for our blue-collar jobs. They can do the tough and repetitive tasks without the need for breaks and do these tasks more efficiently than most humans ever could. Though the predicted result of automation is fairly hard to pin down, the clearest conclusion is that automation will force the original workforce to adapt. When self-driving cars come, society will need someone to decide what the optimal routes are. Regardless of what specific job, automation needs someone to control and tell the robots what to do. The robotic chemist is capable of exposing samples to light and injecting the sample into a gas chromatography instrument, but someone needs to be programming the robot to tell it what they should do if the concentration of their photocatalyst changes. What is the adaptation missing from our current chemist to the robotic chemist? The knowledge is the same and the robot can be built. But the change we need is putting that knowledge into the built robot. And that is why we need to teach our undergraduate STEM students to code.

So if automation is coming for STEM work as well, shouldn't undergraduate STEM students begin changing the priorities of what they are learning as well? I'm specifically talking about science majors such as those focused on physics, chemistry, or engineering disciplines (STEM majors that aren't about computer science). When I was an undergraduate student, I received a very rudimentary introduction to programming through a single class. I learned enough Matlab to make plots and use for loops. And that's enough when you're trying to obtain the varying concentrations of ethanol and water in a distillation column. But as a PhD student, I'm trying to use LabView and Python to more quickly automate my data analysis and collection. I'm attempting to have my data analysis tools communicate with the various instruments I utilize on a daily basis. Most importantly, I'm trying to reduce the time I have to spend performing routine functions so I can think and read more deeply about my research topic. With a better computer science background, this could have gone more smoothly.

I'm not saying we need to drop core classes so that all STEM students can be accomplished AI specialists. But professors and departments should consider updating their curriculum to include more extensive programming classes. I was recently a teaching assistant for the chemical engineering department's hands-on laboratory class. It's a vital class for students to get a feel for how an actual day of an engineer can look like, and a chance for them to work with machinery that they have previously only encountered in their textbooks.

But it was clear to me that the students could leisurely spend their lab time changing parameters with little need for thought or attention. Most groups would assign a person to a single task and make idle talk while the experiment was running. The real trouble came when they had to do analysis outside of lab time, and students were asking me for help in modeling their heat transport or graphing their McCabe-Thiele plots. They're bright students, and eventually were able to figure out how to do all of this with some helpful nudges from me or resources online. But a clear takeaway for me was that this hands-on lab time was fairly unnecessary.

At the end of that class, I didn't want the most important takeaway for students to be their ability to inject a sample from a distillation column stage into a gas chromatography instrument. I wanted it to be their ability to analyze that data and come to conclusions on how best to operate that column. They needed the theory to even begin to decide how they optimize their design procedure. But the coding element was not only crucial to their decision process, but was also the point where most of my students were struggling. Often times, their best resources were their teaching assistants if available, the passed down programs of their upperclassmen, or some obscure forum posts. In 2040, our educators need to create better resources to help students master the coding basics.

We can best prepare these students by giving them better computer science classes. Universities will tend to have separate computer science classes for STEM students. My undergraduate institution did. But why do we need to separate them? There are strong computer science departments and professors in the same university: why not have STEM undergraduates take those same classes? We can reinforce this core lesson by having students continue to flex and exercise those skills in their higher level courses. More theory based classes can place much heavier emphasis on the importance of modeling and coupling their theory with simulations. Professors can assign homework that encourages students to use solvers in addition to their notebook or blackboard calculations. Just like teachers in elementary and middle schools are intertwining their more conventional lessons with new technology, professors can modify their classes similarly.

Robots who can work harder than the most dedicated PhD students are coming. So isn't it time we teach our undergraduates to work smarter, not harder?

[1] Burger, B., Maffettone, P.M., Gusev, V.V. *et al.* A mobile robotic chemist. *Nature* **583,** 237–241 (2020). https://doi.org/10.1038/s41586-020-2442-2

[2] Vincent, James. "Robotic Lab Assistant Is 1,000 Times Faster at Conducting Research." *The Verge*, The Verge, 8 July 2020. https://www.theverge.com/21317052/mobile-autonomous-robot-lab-assistant-research-speed.