Wright Scholar Essay (Topics 1 and 3)

294 Squiggly red underlines. Nearly every line of my code had errors. Null pointers, incompatible types, undefined variables, Gradle sync errors—I had encountered them all. It was February 2024, my freshman year, and we had ten minutes to take the field for our First Tech Challenge (FTC) League Finals. My heart pounded as keys clattered beneath my flying fingers. My code was broken, and for the finals, it had to work. The merciless clock ticked away, and with seconds to go, I finally compiled the code. There was no time to test, hardly any to breathe. We took the field, and my finger hovered over the play button. Time paused. The buzzer sounded, and I pressed play. Success. In two minutes and thirty seconds, we won.

Seven months earlier, I didn't know what a variable was. I was fully into music, and programming wasn't even on my radar. When my friend started a robotics team, I joined on a whim. My journey began with a Google search. Progress was painstakingly slow; it took me two full months to make a motor turn. But gradually, I became hooked. Like a sponge, I absorbed everything: tutorials, documentation, and even Stack Overflow threads. Eventually, I taught myself enough Java to become a functional FTC programmer.

As the season progressed, we became a competitive team, and my knowledge was expanding in parallel. On that competition day, something just clicked. The joy I experienced wasn't just from our robot picking up and scoring pixels, but from seeing my code produce tangible results. In that moment, I'd found my calling. I was no longer just a high school student; I was a STEM student, and I was ready to see where my code could take me.

But that readiness was tested in September 2024. Somewhat naively, I committed to building a machine learning model to predict gait patterns in Parkinson's Disease for my sophomore-year Science Fair project. The problem? I had no clue how. So I dove in: Python syntax, NumPy arrays, signal filtering, feature extraction, and model architectures. I had entered unfamiliar territory, and each concept brought new confusion. After two months of relentless reading, coding, and debugging, I managed to transform raw sensor data into a working classification model. Somewhere between the first error message and the final 96% accuracy, I had begun to absorb a new discipline.

I could have stopped there, but I realized that a working model on my laptop wasn't going to help any Parkinson's patients, and I needed to embed my model into a complete hardware device. This task was beyond daunting, as I had to venture into the foreign territory of hardware and electrical engineering. With my engineering teacher guiding me, I eventually learned the basics. After countless 2 AM KiCAD tutorial sessions, I finally had a working design for a custom printed circuit board (PCB) housing a sensor and microcontroller. Two weeks later, my PCB arrived, and after soldering all my components, it didn't work. My heart sank. In desperation, I resoldered each joint carefully and tried again. Somehow, it worked. After writing some C++ software for the device, I had something that actually worked. The project eventually made it to the International Science and Engineering Fair (ISEF), placing 3rd in Robotics and Intelligent Machines. What struck me most wasn't the placement, but the fact that six months earlier, I wouldn't have understood any of it.

Throughout high school, I've taught myself disciplines, from Java programming to machine learning to circuit design. The Wright Scholar program offers an opportunity to apply my knowledge to critical research. I'm drawn to AFRL's Sensors Directorate, where I hope to deepen my understanding of signal processing while contributing to sensor exploitation technologies. I'm equally fascinated by the Human Performance Wing's work with multimodal sensing to monitor and enhance human performance. What excites me most isn't just the cutting-edge technology, but the chance to work alongside domain experts who can accelerate my growth as an engineer and developer. Whether working with sensor fusion or biomedical sensing, as a sponge eager to learn, AFRL is exactly where I need to be.