I updated my CPSC 326 *mypl* programming language project using DevOps principles. The goal was to create a CI/CD pipeline that would build, test, and lint my project, and then create a .deb package. Once installed, this package would allow users to execute any *mypl* program simply by running *mypl* in the terminal.

1. I used GitHub Actions and a self-hosted runner to implement the CI/CD pipeline. The runner is a Google Compute VM named bobbuilder, and the workflow runs inside a Docker container. I also used Debian packaging to bundle the project and reorganized the directory structure for a cleaner layout that integrates better with the packaging system. Additionally, I wrote a Bash script to create the necessary Debian directories and control files.
2. For building the package, I decided that the executable for *mypl* should be placed in /usr/bin, and the module files should go in /usr/lib/python3/dist-packages/. To support this structure, I reworked some of the import statements in the code. The Bash script handles directory creation and file copying based on this layout.
3. One challenge I faced was figuring out how to properly organize the files, especially since the original repository had everything in a single directory. I also had to learn how Python-specific parts of the Debian directory structure are expected to work.
4. In the future, I plan to set up at least a test harness that runs on each commit so I can ensure that the code I’m pushing is functional. I also intend to continue using Debian packaging, as it provides a much cleaner way to distribute projects.
5. My main takeaway from this class has been all the Linux knowledge I've gained. Before this class, I only used Linux with a few basic commands, I didn’t really understand its elegance. Now, I have a much better appreciation for Linux and a deeper understanding of the development process and what iterating on releases really means.