

5060 Ledge Lane
Williamsville, NY 14221
☎ (716) 697 5190
✉ tjliebel@gmail.com
🌐 troyliebel.me

Troy Liebel

Education

- 2013–2017 **BS, Computer Science**, *University at Buffalo*, Provost Scholarship.
- Software Quality, Software Engineering, Operating Systems, Machine Learning, Modern Networking, Programming Languages, Algorithms, Data Structures.

Work Experience

- 2016–2017 **Computer Science Teaching Assistant**, *University at Buffalo*.
- CSE 442 - Software Engineering, *Fall 2016*:
 - simulated an **Agile/Scrum** work environment for 20 students in four teams via weekly meetings and sprints as they developed a product over 15 weeks.
 - enforced **git** best practices and branching habits for all teams and members.
 - assisted teams in finding useful frameworks, services, and tools.
 - helped with team dynamics and relationships in difficult situations.
 - CSE 115 - Intro to Computer Science, *Spring 2017* - **Java, Python**:
 - established **object oriented programming** concepts for 30 students via weekly lab sessions.
 - prepared 50+ students for future Computer Science courses by hosting three hours of **Python** tutorials each week that served as an introduction to a new language as well as **Unix command line operations** and **Vim**.
- Oct 2014– **Advanced Repair Agent**, *Geek Squad, Best Buy*.
- Present Achieved Quarterly MVP and Employee of the Month on multiple occasions by consistently working to keep average turn around time on client units between 24-48 hours as well as maintaining clear communication with clients through out the repair process, Platinum Certification.

Projects

- Nov 2016 **UB Hacking Finalist**, *Spinbot, C, Python*.
Finished top five of more than 50 teams with a bluetooth connected persistence of vision display created using recycled optical drives, LED's, and Arduinos that featured live message updates via a custom Slack integration.
- Jan 2017 – **Operating Systems**, *os161, ops-class.org*, **C**.
- May 2017 Developed a **Linux Kernel** that can maintain user data across multiple processes by designing and implementing synchronization primitives (locks, semaphores), file system interface (file handle structure, open, close, read, write, lseek), and process and multiprocess support (process structure, fork, execv, waitpid).

Languages, Tools, and Skills

- Fluent C/C++, Python, Java, Git, Atom, OOP, Linux Development Environment.
- Familiar Numpy/Scipy, GoLang, R, Bash, yaml, Erlang, Hadoop, Vim, Eclipse.