

Keith Skinner

SF, Bay Area • (510) 962-2585 • keithskinner93@gmail.com

linkedin.com/in/keith-skinner 
github.com/keith-skinner 

Summary

A full stack software engineer dedicated to organized, performant code. With experience in developing, testing, and designing in multiple programming paradigms and languages. Using up to date and relevant industry standards, principals, and best practices, I am able to produce fast, reliable, and maintainable software.

Professional Skills

Languages: C/C++, C#, Java, Python, Bash, SQL

Tools: Git, Clang, CMake, vxWorks, Jenkins, IBM DOORS, VectorCAST, Visual Studio, Coverity, Bitbucket, Confluence, Protocol Buffers, Boost Libraries, TestComplete, GDB, Valgrind, MySQL, Node, ReactJS, SCSS

Work Experience

Software Engineer

ArgonST, Inc. A Boeing Company

Sept. 2020 - Present

Impact

- Recreated many C++ STL containers, allocators, and related classes due to program requirements restricting the use of exceptions and allocations, as well as, lack of constexpr support in **C++14**.
- Decoupled the embedded GUI from the windows GUI implementation to reduce duplicated **OpenGL** code.
- Created test application for non-developers to simulate poor communication conditions including latency and message corruption in **C#**.
- Created Jenkins pipeline to enforce coding requirements and output/reject non-compliant code sections (**Jenkins, Bash, Python, Coverity**).
- Created **CMake** scripts that moved runtime operations into build-time constructs which were then able to be copied into libraries and executables.
- Fixed Squashing of data being presented to tables in longstanding **Java** table implementation.
- Created a source code parsing script that updated or created copyright-comments for new and previously updated files using **Python** and **Bash** utilities.

Responsibilities

- Process, interpret and display data relating to radio frequency technologies
- Develop new features and tools to improve and support existing products in C++, C#, Java, and Python
- Implement integration and end-to-end testing scripts with internal and commercial tools
- Decouple embedded and desktop dependencies to allow for desktop emulation
- Implement design specifications and requirements into concrete features
- Refactor major system components in a legacy code base
- Leverage debuggers and unit tests to confirm behavior and correctness
- Document system behavior, implementation, and intent

Education

CSU Channel Islands

Bachelor of Science Computer Science

Camarillo, CA
Dec. 2019

Projects

Sharp 35902 SoC Emulator

Simulates a Sharp 35902 CPU with C++23. A single-threaded CPU and part of a SoC with memory mapped registers, interrupts, bank switched ROM, RAM, and tile map.