Low-level programming requires careful management of system resources, most notably memory. In C++ programmers are encouraged to follow idioms like RAII and smart pointers to handle resources correctly as violating them leads to unsafe code.

Typed functional programming languages guarantee safe automatic memory management, but are often sub-optimal in handling system resources. A nice, formal solution to handling resources naturally is linear types. Unfortunately, existing languages that support linearity are cumbersome and require explicit, complicated annotations from the programmer.

We bridge the two worlds by exploring a novel combination of C++ and linear types. We describe a new type system with linearity for C++ by using constrained qualified types, while requiring no additional input from the programmer. The applied result of our work is called Lily, a static analysis tool for C++ using the Clang compiler infrastructure. Lily can statically detect large, general classes of issues, some of which are not detected by common state-of-the-art tools for C++.