

Document	P1732R2
Date	2019-12-10
Author	CJ Johnson < <a href="mailto:johnsoncj@google.com">johnsoncj@google.com</a> >
Audience	Library Evolution Working Group (LEWG), Library Working Group (LWG)

# Do not promise support for function syntax of operators

---

From Hyrum's Law [1]:

With a sufficient number of users of an API,  
it does not matter what you promise in the contract:  
all observable behaviors of your system  
will be depended on by somebody.

## Updates over Revision 1

- Based on discussion in Belfast, switched to a less restrictive wording that permits, as an example, function-style invocation of `operator->()`.

## Updates over Revision 0

- Removed extra wording options, settling on the only remaining proposed option, thanks to input from Walter Brown.

## Proposal

In the Library Evolution Working Group (LEWG) room of the Kona 2019 ISO C++ meeting, it was mentioned that the Standard Library reserves the right to change the way operators are implemented (switching between member and non-member overloading). This paper proposes updating SD-8 to convey this information to users.

For clarity...

Subject to breakage: `x.operator+(y)`, `operator+(x, y)`

Will not break: `x + y`

## Wording

Primarily, the standard reserves the right to:

[...]

+ \* Change the implementation of overloaded operators between member and non-member style.

[...]

## References

[1] [Hyrum's Law](#)