Document	P1732R4
Date	2020-02-21
Author	CJ Johnson <johnsoncj@google.com>
Audience	Library Evolution Working Group (LEWG)

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 3

- Further wording clarification

Updates over Revision 2

- Corrected paper audience
- Based on discussion in Prague, replaced proposed wording to the final, agreed upon version.

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.

Wording

```
  Primarily, the standard reserves the right to:

  [...]

+ * Assume operators are invoked using only operator syntax (a + b), not function syntax
+   (a.operator+(b) or operator+(a, b)). An exception is made for operators that can be
+   overloaded with member syntax (operator=, operator->, etc) only.

  [...]
```

References

[1] Hyrum's Law