The Compute Method

```
def compute(self, n: int) -> int:
    if n == 0: return 1
    return n*self.__call__(n-1)
```

- The real work of the callable object.
- A pretty standard recursive factorial definition.
- Depends on memoization to have previous results.

Note the issue

- Hysteresis — memory of what came before
  - A callable object can be stateful
- This may be desirable to optimize performance
- It may be confusing if done badly

We generally expect idempotence:
functions do the same thing each time