`std::weak_ptr`

`std::weak_ptr` isn't really a smart pointer.

- `std::weak_ptr`
  - owns no resource.
  - shares the resource with a `std::shared_ptr`.
  - supports no transparent access on the resource.

The `std::weak_ptr` does not change the reference counter.

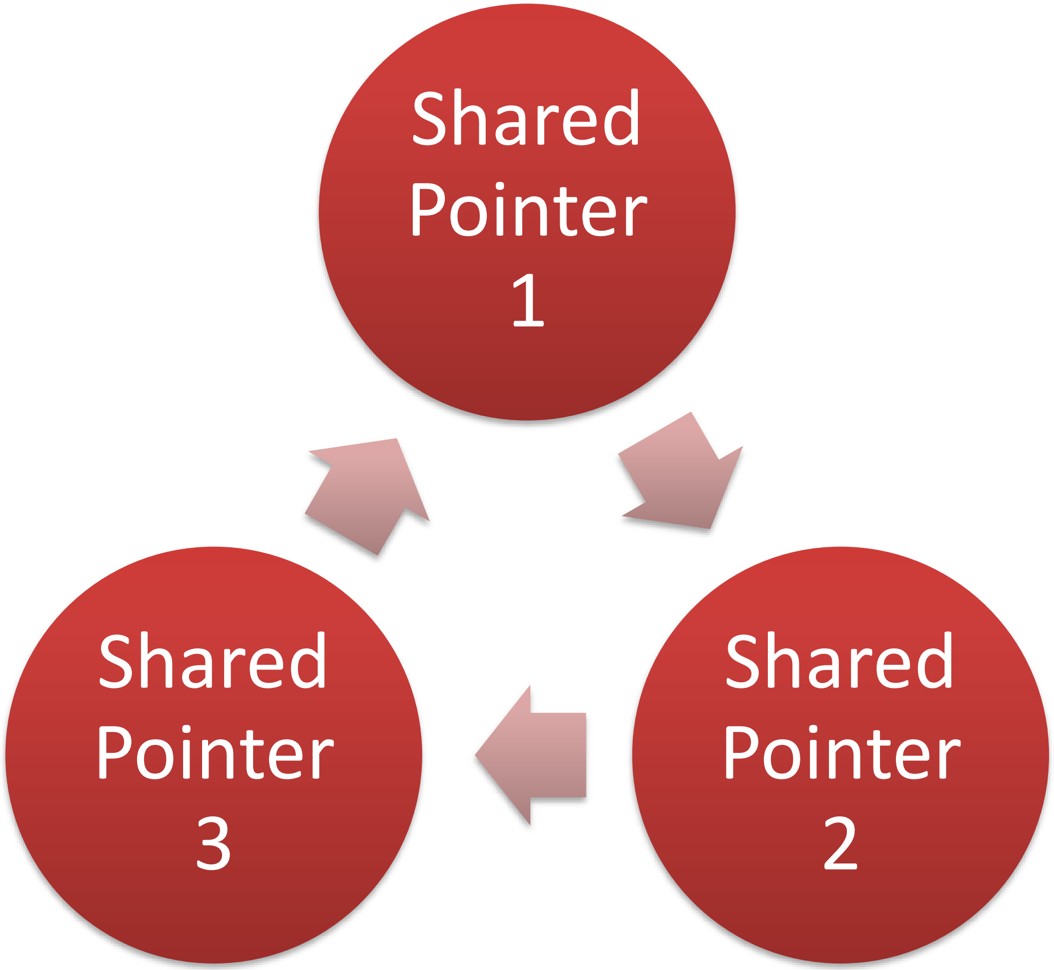
➡ His purpose is it to break cyclic references of `std::shared_ptr`.

std::weak_ptr

Member Function	Description
`wea.expired()`	Checks, if the resource was already deleted.
`wea.use_count()`	Returns the value of the reference counter.
`wea.lock()`	Generates a `std::shared_ptr` from the `std::weak_ptr.`
`wea.reset()`	Releases the resource.
`wea.swap(wea2)`	Swaps `std::weak_ptr`

weakPtr.cpp

Cyclic References



- Classical problem:

  `std::shared_ptr` creates a cycle, so that no resource can be releases automatically.

- Rescue:

  `std::weak_ptr` breaks the cycles of `std::shared_ptr`.