

# Smart Pointers: Weak Pointers

`std::weak_ptr` is the last component of the smart pointers family. Its purpose is limited compared to the other smart pointer, and we will examine why in this lesson.

## WE'LL COVER THE FOLLOWING ^

- Introduction
- Methods

## Introduction #

To be honest, `std::weak_ptr` is not a classic smart pointer. `std::weak_ptr` supports no transparent access to the resource as it only borrows the resource from a `std::shared_ptr`.

## Methods #

The table provides an overview of the methods of `std::weak_ptr`.

Name	Description
<code>expired</code>	Checks if the resource was deleted.
<code>lock</code>	Creates a <code>std::shared_ptr</code> on the resource.
<code>reset</code>	Resets the resource.
<code>swap</code>	Swaps the resources.
<code>use_count</code>	Returns the value of the reference counter.

counter.

## Methods of `std::weak_ptr`



There is one main reason for the existence and use of `std::weak_ptr`. It breaks the cycle of `std::shared_ptr`. We will discuss these [cyclic references](#) in detail in the next lessons.

---

Let's see an example of this topic in the next lesson.