# CRTP

Let's learn about CRTP in this lesson.

WE'LL COVER THE FOLLOWING



## CRTP

The acronym CRTP stands for the C++ idiom **C**uriously **R**ecurring **T**emplate **P**attern and is a technique in C++ in which a `Derived` class derives from a class template `Base`. The key is that `Base` has `Derived` as a template argument.

Let’s have a look at an example:

```
template<class T>
class Base{
   ...
};

class Derived: public Base<Derived>{
   ...
};
```

> CRTP enables static polymorphism.

## Typical use-case

There are two typical use-cases for CRTP: Mixins and static polymorphism.

## Mixins

Mixins is a popular concept in the design of classes to mix in new code. Therefore, it's an often-used technique in Python to change the behavior of a class by using multiple inheritances. In contrast to C++, in Python, it is legal to have more than one definition of a method in a class hierarchy. Python simply uses the method that is first in the Method Resolution Order (MRO).

You can implement mixins in C++ by using CRTP. A prominent example is the class `std::enable_shared_from_this`. By using this class, you can create objects that return an `std::shared_ptr` to themselves. We have to derive your class `MySharedClass` public from `std::enable_shared_from_this`. Now, our class `MySharedClass` has a method `shared_from_this`.

An additional typical use-case for mixins is a class that you want to extend with the capability that their instances support the comparison for equality and inequality.

## Static Polymorphism

Static polymorphism is quite similar to dynamic polymorphism. But contrary to dynamic polymorphism with virtual methods, the dispatch of the method calls will take place at compile-time. Now, we are at the center of the CRTP idiom.

```
class ShareMe: public std::enable_shared_from_this<ShareMe>{
  std::shared_ptr<ShareMe> getShared(){
    return shared_from_this();
  }
};
```

- `std::enable_shared_from_this` creates a `shared _ptr` for an object.
- `std::enable_shared_from_this:` base class of the object.
- `shared_from_this:` returns the shared object

To learn more about CRTP, click here.

---

In the next lesson, we'll look at a couple of examples of CRTP.