

Assignment 6: CRTP

C++ Programming Course, Summer Term 2018

6-0 References and Prerequisites

We discussed the basic principle of the Curiously Recurring Template Pattern (CRTP).

Experiment with this [minimal example](#), make sure you understand the type-related mechanisms behind this technique.

There are good references on CRTP around (also see [session 6](#)), the explanations in [this video](#) should help you a lot. Again [compiler explorer](#) and [CppInsights](#) are very useful.

6-1 CRTP: Iterator Base

6-1-1 Reference Implementation

Study the `IteratorBase` class template in DASH:

- [IteratorBase](#)
 - Note that it is subclassing `std::iterator` - don't do this. This is because DASH is an STL implementation so we wanted our iterators to type-match STL iterators, but this is deprecated style (see note [N3931 on open-std.org](#))
 - Default constructor is deleted to forbid direct instantiation of `IteratorBase`. Derived iterator classes should be default-constructible.
- [Usage of IteratorBase](#)

Search for `IndexSetIterator`. It is not default-constructible for rather peculiar reasons, don't imitate this.

6-1-2 Iterator Base Classes for `list` and `sparse_array`

Implement CRTP iterator base class templates:

- `RandomAccessIteratorBase<...>`
- `ForwardIteratorBase<...>`

Use them as base class of the iterators in your implementation of `list<T,x>` and `sparse_array<T,N>`. Test cases do not have to be extended but should pass.