The Curiously Recurring Template Pattern (CRTP)

William Oakley

April 12, 2019

What is CRTP?

- The Curiously Recurring Template Pattern (CRTP) is a *programming idiom* in C++ in which a class X derives from a class template instantiation using X itself as a template argument:

```
template <class T>
class Base {
    // stuff
};
class Derived : public Base<Derived> {
    //stuff
};
```

- It is also called "upsidedown inheritence" since CRTP extends a class hierarchy "upwards" rather than "downwards."

Example of CRTP

Object Counter - Keep track of number of instances of an object.

Is CRTP Used?

- Used all over the Microsoft Active Template Library (ATL).
- Used by the clang compiler frontend.
- Used by Boost for its iterator facade.

How/when is CRTP Useful?

- Extending/customizing the functionality of a class *in a manner that is specific to the class* while retaining a common interface (polymorphism).
- *Dynamic* polymorphism is not required.
- $\therefore$ *Static* polymorphism with CRTP.

Polymorphism

- **Polymorphism**: Provision of a single interface to entities of different types.
- **Dynamic Polymorphism**: Implementation details of the interface are determined at runtime.
  - Ex: User input required, runtime randomness
  - Virtual functions
- **Static Polymorphism**: Implementation details of the interface are determined at compile-time.
- Ex: All inputs are determined at compile-time.
- Templates

Real-world Example of CRTP

Visitor Class

- Used by clang.

Performance Testing of CRTP

Counter example.

Performance Testing of CRTP

- Using static polymorphism rather than dynamic polymorphism allows the compiler to (perhaps greatly) optimize your code.
- The key optimization is inlining.
  - Inlining removes function calls.
  - Inlining increases the code size of functions allowing for more extensive compiler optimizations.
- Inlining can decrease performance if the hot section of code does not fit in one level of the memory heirarchy after expansion.

Bonus Topic: Mixins

Point example.

References

1. https://www.fluentcpp.com/2018/05/22/how-to-transform-a-hierarchy-of-virtual-methods-into-a-crtp
2. https://eli.thegreenplace.net/2011/05/17/the-curiously-recurring-template-pattern-in-c
3. https://en.wikipedia.org/wiki/Curiously_recurring_template_patternj
4. Vandevoorde, David, Nicolai M. Josuttis, and Douglas Gregor. C++ Templates: The Complete Guide. 2018.