
Subject: Re: Nested template question

Posted by [Novo](#) on Mon, 10 Jun 2019 18:42:55 GMT

[View Forum Message](#) <> [Reply to Message](#)

```
struct AZero {  
    template <typename T>  
    operator T() const {return 0;}  
  
    template <typename T>  
    operator std::complex<T>() const {return std::complex<T>(0, 0);}  
};
```

```
double val    = AZero();  
std::complex<float> valc1 = AZero();  
std::complex<double> valc2 = AZero();  
Checked with Clang.
```

What you are trying to do is

```
template <typename T> T GetAZero() {return 0;}  
template <typename T> std::complex<T> GetAZero<class std::complex<T>>() {return  
std::complex<T> (0, 0);}
```

It won't compile because this is partial function specialization, which is allowed only for classes.

The way I implemented this is also partial function specialization, but for some reason it compiles
:roll:
