

Forum:[Hack Forums](#)

Thread: [Overloading operator << of nested class where enclosing class is a template](#)

lockhead - April 22, 2021, 7:10 am

Hi guys,

I'm trying to overload the << operator for a nested class that lives inside an enclosing class which is templated.

The enclosing class is my attempt at implementing a std::vector, the nested class is an iterator.

The way I understood it, when overloading the << operator you are actually overloading the operator of the ostream object, not of your class, therefore you cannot implement it in your class. You must instead declare the operator in your class as a "friend" and implement it within the global scope.

Pseudo Code:

PHP Code

```
<?php
template <typename T>
class Vec{
public:
    class iterator : std::iterator<...>{
        ...
        friend std::ostream& operator<<(const std::ostream& os, const Vec<T>::iterator& it);
    }
}

/*global scope */
template <typename T>
std::ostream& operator<<(const std::ostream& os, const Vec<T>::iterator& it){
    os << it->num;
    return os;
}
```

[Link to Full Code in Pastebin](#)

The errors I am getting when compiling with VisualStudio

Code use the 'typename' keyword to treat nontype "Vec<T>::iterator [with T=T]" as a type in a dependent context

'Vec<T>::iterator': dependent name is not a type

syntax error: identifier iterator

binary 'operator <<' has too few parameters

The thing I am struggling with is correctly typing the parameters for the overload function. I've tried

many things but can't get it to work, or find relevant answers online, so any help would be greatly appreciated!