

---

Subject: Template problem

Posted by [bytefield](#) on Wed, 20 Feb 2008 21:54:44 GMT

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

---

Hi. I have one example about how to create a "safe" vector inheriting it from STL vector. The code is showed below. I have errors compiling it under Ubuntu.

```
#include <vector>
#include <iostream>
#include <string>
using namespace std;

struct Entry
{
    string name;
    int number;
};

template< class T > class Vec: public vector< T >
{
public:
    Vec(): vector< T >(){}
    Vec(int s): vector< T >(s){}

    T& operator[](int i) { return vector<T>::at(i); }
    const T& operator[](int i) const { return vector<T>::at(i); } // oups, forgotten to put last const
    keyword
};

Vec< Entry > tel(1000);

void print_entry(int i)
{
    cout << tel[i].name << ' ' << tel[i].number << '\n';
}

int main()
{
    try
    {
        print_entry(1000);
    }
    catch(out_of_range)
    {
        std::cerr << "Domain error\n";
        return 1;
    }
    catch(...)
```

```
{
  std::cerr << "Unknow exception\n";
  return 1;
}
return 0;
}
```

What could be the problem(s)?

Compiler version:

Using built-in specs.

Target: i486-linux-gnu

Configured with: ../src/configure -v --enable-languages=c,c++,fortran,objc,obj-c++,treelang  
--prefix=/usr --enable-shared --with-system-zlib --libexecdir=/usr/lib --without-included-gettext  
--enable-threads=posix --enable-nls --with-gxx-include-dir=/usr/include/c++/4.1.3  
--program-suffix=-4.1 --enable-\_\_cxa\_atexit --enable-clocale=gnu --enable-libstdcxx-debug  
--enable-mpfr --enable-checking=release i486-linux-gnu

Thread model: posix

gcc version 4.1.3 20070929 (prerelease) (Ubuntu 4.1.2-16ubuntu2)

Errors:

vec.cpp:22: instantiated from here

L.E: See line 19 `const T& operator[](int i) const { return vector<T>::at(i); }`  
My mistake, now overloading the `[]` operator work.

---