

Re: Class templates and friend function templates

Source: <http://www.tech-archive.net/Archive/VC/microsoft.public.vc.language/2005-07/msg01066.html>

- *From:* "Carl Daniel [VC++ MVP]" <cpdaniel_remove_this_and_nospam@xxxxxxxxxxxxxxxxxx>
 - *Date:* Sat, 23 Jul 2005 15:50:56 -0700
-

BigMan wrote:

> I have a class template like this:

>

> template< typename t >

> class c;

>

> I'd also like to have an operator == for objects of types, which are

> different specializations of c:

> template< typename t1, typename t2 >

> bool operator ==

> (

> s< t1 > const&

> , s< t2 > const&

>);

>

> In order to implement this operator, I need it to be a friend of both

> s< t1 > and s< t2 >. For the sake of safety, I'd like no other

> specialization of c to have this operator as its friend. Is this

> possible and, if so, how?

Like this:

```
// Code
```

```
template <class T> class C;
```

```
template <class U, class V> bool operator ==(C<U>,C<V>);
```

```
template <class T> class C
```

```
{
```

```
public: C(int x_) : x(x_) {}
```

```
private: int x;
```

```
template <class U,class V>
```

```
friend bool operator == <T>(U u, V v);
```

```
};
```

```
template <class U, class V>
```

```
bool operator == (C<U> u, C<V> v)
```

Re: Class templates and friend function templates

```
{  
return u.x == v.x;  
}
```

```
int main()  
{  
C<int> c1(1);  
C<float> c2(2);  
bool b = c1 == c2;  
}  
// End of code
```

I'm actually not sure that this code is standard conforming, although it seems like it should be. VC 7.1 likes it, but Comeau claims that the template friend declaration is ill-formed.

-cd

- *Follow-Ups:*

- ◆ *Re: Class templates and friend function templates*

◇ From: Carl Daniel [VC++ MVP]

- *References:*

- ◆ *Class templates and friend function templates*

◇ From: BigMan

- Prev by Date: *cast array to structure*
- Next by Date: *Re: How to set a backgroundcolor for a windows control (Trackbar) ?*
- Previous by thread: *Class templates and friend function templates*
- Next by thread: *Re: Class templates and friend function templates*
- Index(es):
 - ◆ *Date*
 - ◆ *Thread*