

# Discuss C++ Template Downcast

## Table of Contents

- Original Discuss
- The problem
- My answer to the problem
  - First we discuss ff
  - Then we discuss f

This is a discuss in C board in [bbs.sjtu.edu.cn](http://bbs.sjtu.edu.cn), about type down-cast in C++ template.

## Original Discuss

<http://bbs.sjtu.edu.cn/bbstdcon/board/C/thread/13300789>

# The problem

Today I read a book about we can do cast-down in template, so I write this to test:

```
1  τεμπλατε <βοολ _Τεστ, χλασσ _Τψπε = ωοιδ>
2  στρυχτ εναβλε_ιφ { };
3
4  τεμπλατε<χλασσ _Τψπε>
5  στρυχτ εναβλε_ιφ<τρυε, _Τψπε> {
6      τψπεδεφ _Τψπε τψπε;
7  };
8
9  χλασσ A { };
10 χλασσ B : A { };
11
12 τεμπλατε <τψπεναμε T>
13 στρυχτ τραιτσ { στατιχ ιντ χονστ ωαλυε = φαλσε; };
14
15 τεμπλατε <>
16 στρυχτ τραιτσ<A> { στατιχ ιντ χονστ ωαλυε = τρυε; };
17
18 τεμπλατε <τψπεναμε T>
19 ωοιδ φ(T, τψπεναμε εναβλε_ιφ<τραιτσ<T>::ωαλυε>::τψπε* = 0
20
21 τεμπλατε <>
22 ωοιδ φ<A>(A, εναβλε_ιφ<τραιτσ<A>::ωαλυε>::τψπε*) { }
23
24
25
26 τεμπλατε <τψπεναμε T>
```

```

27  χλασσ BB {};
28
29  τεμπλατε <τυπεναμε T>
30  χλασσ ΔΔ : πυβλιχ BB<T> {};
31
32  τεμπλατε <τυπεναμε T> ποιδ φφ(BB<T>) {};
33
34  ιντ μαιν(ιντ αργχ, χηαρ * αργω[])
35  {
36      Α α; Β β;
37      ΔΔ<λονγ> δδ;
38      //φ(β);
39      φφ(δδ);
40  }

```

It is strange when  $\phi$  it don't allow my specified  $\phi<A>$  <sup>-</sup>.

But in  $\phi\phi$  it allowed  $\phi\phi<BB<λονγ>>$  <sup>-</sup>.

Tested under VC10 and GCC3.4

## My answer to the problem

Let's think ourself as compiler to see what happened there.

Define mark # : **A#B** is the instantiated result when we put **B** into the parameter **T** of **A<T>** .

# First we discuss ff

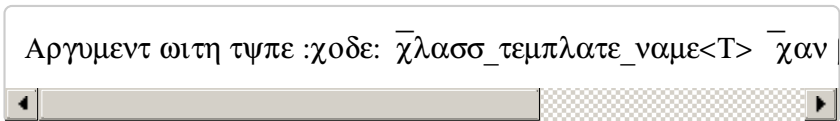
```
1 ΔΔ<λονγ> δδ;
```

After this sentence, the compiler saw the instantiation of `ΔΔ<λονγ>` , so it instantiate `ΔΔ#λονγ` , and also `BB#λονγ` .

```
1 φφ(δδ);
```

This sentence required the compiler to calculate set of overloading functions.

Step 1 we need to infer `T` of `φφ<T>` from argument `ΔΔ#λονγ -> BB<T>` . Based on the inference rule:



So compiler inferred `T` as `λονγ` . Here if it is not `BB` but `XX` which is complete un-related, we can also infer, as long as `XX` is a template like `XX<T>` .

Step 2 Template Specialization Resolution. There is only one template here so we matched `φφ<T>` .

Step 3 Template Instantiation

After inferred `λονγ -> T` , compiler instantiated `φφ#λονγ` .

Set of available overloading functions : `{φφ#λονγ}`

Then overloading resolution found the only match

$\phi\#\lambda\text{ον}\gamma^-$ , checked its real parameter  $\Delta\Delta\#\lambda\text{ον}\gamma$  can be down-cast to formal parameter  $BB\#\lambda\text{ον}\gamma$ .

## Then we discuss f

1  $\phi(\beta)$ ;

Calculate set of overloading functions.

Step 1 infer all template parameters for template  $\phi$ . According to inference rule:

Παραμετερ ωιτη τυπε T χαν βε υσεδ το ινφερ T

So  $B \rightarrow T$  is inferred.

Step 2 Template Specialization Resolution.

Here  $B$  is not  $A$  so we can not apply specialization of  $\phi\langle A \rangle$ , remaining  $\phi\langle T \rangle$  as the only alternative.

Step 3 Template Instantiation.

When we put  $B$  into  $\phi\langle T \rangle$  to instantiate as  $\phi\#B$ , we need to instantiate  $\text{τραιτ}\phi\#B^-$ .

There is no specialization for  $B$  so we use template  $\text{τραιτ}\phi\langle T \rangle$ ,  $\text{τραιτ}\phi\#B::\omega\alpha\lambda\upsilon\epsilon=\phi\alpha\lambda\varsigma\epsilon$ , so  $\epsilon\nu\alpha\beta\lambda\epsilon\_i\phi\#\phi\alpha\lambda\varsigma\epsilon$  didn't contains a  $\text{τυπε}$ , an error occurred.

The only template is mismatch, available overloading functions is empty set. So we got an error.

