C++ final specifier

Difficulty Level : Easy ● Last Updated : 03 Nov, 2020

In Java, we can use final for a function to make sure that it cannot be overridden. We can also use final in Java to make sure that a class cannot be inherited. Similarly, the latest C++ standard C++ 11 added final.

Use of final specifier in C++ 11:

Sometimes you don't want to allow derived class to override the base class' virtual function. C++ 11 allows built-in facility to prevent overriding of virtual function using final specifier.

Consider the following example which shows use of final specifier. This program fails in compilation.

CPP

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

class Base
{
public:
    virtual void myfun() final
    {
        cout << "myfun() in Base";
    }
};
class Derived : public Base
{
    void myfun()
    {
        cout << "myfun() in Derived\n";
    }
};

int main()
{
    Derived d;
    Base &b = d;
```

```
}
```

Output:

```
prog.cpp:14:10: error: virtual function ‘virtual void Derived::myfun()’
     void myfun()
          ^
prog.cpp:7:18: error: overriding final function ‘virtual void Base::myf
     virtual void myfun() final
```

2nd use of final specifier:

final specifier in C++ 11 can also be used to prevent inheritance of class / struct. If a class or struct is marked as final then it becomes non inheritable and it cannot be used as base class/struct.

The following program shows use of final specifier to make class non inheritable:

CPP

```
#include <iostream>
class Base final
{
};

class Derived : public Base
{
};

int main()
```

```
}
```

Output:

```
error: cannot derive from ‘final’ base ‘Base’ in derived type ‘Derived’
 class Derived : public Base
```

final in C++ 11 vs in Java

Note that use of final specifier in C++ 11 is same as in Java but Java uses final before the class name while final specifier is used after the class name in C++ 11. Same way Java uses final keyword in the beginning of method definition (Before the return type of method) but C++ 11 uses final specifier after the function name.

CPP

```cpp
class Test
{
    final void fun()// use of final in Java
    { }
}
class Test
{
public:
    virtual void fun() final //use of final in C++ 11
    {}
};
```

Unlike Java, final is not a keyword in C++ 11. final has meaning only when used in above contexts, otherwise it’s just an identifier.
One possible reason to not make final a keyword is to ensure backward compatibility. There may exist production codes which use final for other purposes. For example the following program compiles and runs without error.

CPP

```cpp
#include <iostream>
using namespace std;
```

```
    int final = 20;
    cout << final;
    return 0;
}
```

Output:

```
20
```

In java, final can also be used with variables to make sure that a value can only be assigned once. this use of final is not there in C++ 11.
This article is contributed **Meet Pravasi**. Please write comments if you find anything incorrect, or you want to share more information about the topic discussed above

Want to learn from the best curated videos and practice problems, check out the C++ Foundation Course for Basic to Advanced C++ and C++ STL Course for foundation plus STL. To complete your preparation from learning a language to DS Algo and many more, please refer **Complete Interview Preparation Course**.

Previous Next

RECOMMENDED ARTICLES

Page : 1 2

03 Difference between %d and %i format specifier in C language
14, Jul 17

04 _Noreturn function specifier in C
06, Sep 17

07 Different ways of accessing array elements in C++
17, May 21

08 class std::string_view in C++17
16, May 21

Article Contributed By :

GeeksforGeeks

Vote for difficulty

Current difficulty : Easy

Easy | Normal | Medium | Hard | Expert

Improved By : abhinav bajpai

Article Tags : CPP-Functions, cpp-virtual, C Language, C++

Practice Tags : CPP

Improve Article | Report Issue

Writing code in comment? Please use ide.geeksforgeeks.org, generate link and share the link here.

Load Comments

Sector-136, Noida, Uttar Pradesh - 201305

feedback@geeksforgeeks.org