



# Accessing Inherited Functions



by vatsalchanana

Problem

Submissions

Leaderboard

Discussions

You are given three classes *A*, *B* and *C*. All three classes implement their own version of *func*.

In class *A*, *func* multiplies the value passed as a parameter by **2**:

```
class A
{
    public:
        A(){
            callA = 0;
        }
    private:
        int callA;
        void inc(){
            callA++;
        }

    protected:
        void func(int & a)
        {
            a = a * 2;
            inc();
        }
    public:
        int getA(){
            return callA;
        }
};
```

In class *B*, *func* multiplies the value passed as a parameter by **3**:

```
class B
{
    public:
        B(){
            callB = 0;
        }
    private:
        int callB;
        void inc(){
            callB++;
        }

    protected:
        void func(int & a)
        {
            a = a * 3;
            inc();
        }
    public:
        int getB(){
            return callB;
        }
};
```

In class *C*, *func* multiplies the value passed as a parameter by **5**:

```

class C
{
public:
    C(){
        callC = 0;
    }
private:
    int callC;
    void inc(){
        callC++;
    }
protected:
    void func(int & a)
    {
        a = a * 5;
        inc();
    }
public:
    int getC(){
        return callC;
    }
};

```

You are given a class *D*:

```

class D
{
    int val;
public:
    //Initially val is 1
    D()
    {
        val = 1;
    }

    //Implement this function
    void update_val(int new_val)
    {

    }
    //For Checking Purpose
    void check(int); //Do not delete this line.
};

```

You need to modify the class *D* and implement the function `update_val` which sets *D*'s *val* to *new\_val* by manipulating the value by only calling the *func* defined in classes *A*, *B* and *C*.

It is guaranteed that *new\_val* has only 2, 3 and 5 as its prime factors.

#### Input Format

Implement class *D*'s function `update_val`. This function should update *D*'s *val* only by calling *A*, *B* and *C*'s *func*.

#### Constraints

$1 \leq \text{new\_val} \leq 10000$

**Note:** The *new\_val* only has 2, 3 and 5 as its prime factors.

#### Sample Input

*new\_val* = 30

#### Sample Output

*A*'s *func* will be called once.

*B*'s *func* will be called once.

*C*'s *func* will be called once.

#### Explanation

Initially,  $val = 1$ .

A's *func* is called once:

```
val = val*2  
val = 2
```

B's *func* is called once:

```
val = val*3  
val = 6
```

C's *func* is called once:

```
val = val*5  
val = 30
```

f t in

Submissions: 8474

Max Score: 30



Difficulty: Medium

Rate This Challenge:

☆☆☆☆☆

[More](#)

Need Help? Get advice from the [discussion forum](#) for this challenge. Or check out the [environments page](#)

Current Buffer (saved locally, editable)  

C++



```
1 ▶ #include<↔  
2  
3 using namespace std;  
4  
5 class A  
6 ▼ {  
7     public:  
8     A(){  
9         callA = 0;  
10    }  
11    private:  
12        int callA;  
13    void inc(){  
14        callA++;  
15    }  
16  
17    protected:  
18        void func(int & a)  
19    {  
20        a = a * 2;  
21        inc();  
22    }  
23    public:  
24    int getA(){  
25        return callA;  
26    }  
27 };  
28  
29 class B  
30 ▼ {  
31     public:  
32     B(){  
33         callB = 0;  
34     }
```

```
35     private:
36         int callB;
37     void inc(){
38         callB++;
39     }
40     protected:
41     void func(int & a)
42     {
43         a = a * 3;
44         inc();
45     }
46     public:
47     int getB(){
48         return callB;
49     }
50 };
51
52 class C
53 {
54     public:
55     C(){
56         callC = 0;
57     }
58     private:
59     int callC;
60     void inc(){
61         callC++;
62     }
63     protected:
64     void func(int & a)
65     {
66         a = a * 5;
67         inc();
68     }
69     public:
70     int getC(){
71         return callC;
72     }
73 };
74
75 class D :private A,private B,private C
76 {
77     int val;
78     public:
79         //Initially val is 1
80         D()
81         {
82             val = 1;
83         }
84
85         //Implement this function
86         void update_val(int new_val)
87         {
88             while (new_val%2==0)
89             {
90                 new_val /= 2;
91                 A::func(val);
92             }
93             while (!(new_val%3))
94             {
95                 new_val /= 3;
96                 B::func(val);
97             }
98             while (!(new_val%5))
99             {
100                 new_val /= 5;
101                 C::func(val);
102             }
103         }
104
105         //For Checking Purpose
106 }
```

```
107         void check(int); //Do not delete this line.
108     };
109
110
111
112 void D::check(int new_val)
113 {
114     update_val(new_val);
115     cout << "Value = " << val << endl << "A's func called " << getA() << " times " << endl << "B's func called " <<
getB() << " times" << endl << "C's func called " << getC() << " times" << endl;
116 }
117
118
119 int main()
120 {
121     D d;
122     int new_val;
123     cin >> new_val;
124     d.check(new_val);
125 }
126 }
```

Line: 103 Col: 15

 [Upload Code as File](#)☐ Test against custom input[Run Code](#)[Submit Code](#)

## Congrats, you solved this challenge!

✓ Test Case #0  
✓ Test Case #3  
✓ Test Case #6  
✓ Test Case #9

✓ Test Case #1  
✓ Test Case #4  
✓ Test Case #7

✓ Test Case #2  
✓ Test Case #5  
✓ Test Case #8

[Next Challenge](#)

You've earned 30.00 points!

Copyright © 2017 HackerRank. All Rights Reserved

Join us on IRC at [#hackerrank](#) on freenode for hugs or bugs.[Contest Calendar](#) | [Interview Prep](#) | [Blog](#) | [Scoring](#) | [Environment](#) | [FAQ](#) | [About Us](#) | [Support](#) | [Careers](#) | [Terms Of Service](#) | [Privacy Policy](#) | [Request a Feature](#)