




```
v.~uint32_t();
```

Matthis Kruse

 +  dasnacl  
 matthis-kruse

September 16, 2019

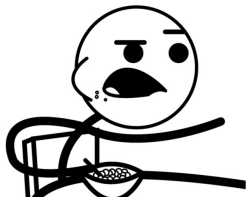
```
int w = 0;  
std::uint32_t v = 0;
```

```
int w = 0;  
std::uint32_t v = 0;  
  
v.~uint32_t();
```

```
int w = 0;
std::uint32_t v = 0;

v.~uint32_t();
// w.~int(); // doesn't compile
```

```
int w = 0;  
std::uint32_t v = 0;  
  
v.~uint32_t();  
// w.~int(); // doesn't compile
```



```
#define ever (;;)
```

```
int main()  
{  
    for ever  
    {  
        fmt::print("hi");  
    }  
}
```

```
const int west_c = -1;
```

```
const int west_c = -1;  
int const east_c = 1;
```



```
typedef int west_t;  
int typedef east_t;
```

**WEST SIDE**



**EAST SIDE**



```
signed typedef wa_t;
```

```
int size = 1729;  
signed typedef wa_t[size];
```

```
int size = 1729;  
signed typedef wa_t[size];
```

```
wa_t even { 2 };  
fmt::print("{} ", sizeof(even) / 4));
```

```
int size = 1729;
signed typedef wa_t[size];
//unsigned typedef wo_t[size]; // doesn't compile

wa_t even { 2 };
fmt::print("{} ", sizeof(even) / 4); // prints '1729'
```

```
int size = 1729;  
signed typedef wa_t[size];  
//unsigned typedef wo_t[size]; // doesn't compile  
  
wa_t even { 2 };  
fmt::print("{} ", sizeof(even) / 4); // prints '1729'
```



```
int i(int(x));
```

```
int i(int(x));  
//fmt::print("{} ", i); // doesn't compile
```



```
int i(int(x));  
//fmt::print("{} ", i); // doesn't compile  
  
int j((int(x)));
```

```
int i(int(x));  
//fmt::print("{} ", i); // doesn't compile
```

```
int j((int(x)));  
fmt::print("{} ", j);
```

```
int i(int(x));  
//fmt::print("{} ", i); // doesn't compile
```

```
int j((int(x)));  
fmt::print("{} ", j);    // ><(((°>
```

```
int i(int(x));  
//fmt::print("{} ", i); // doesn't compile
```

```
int j((int(x)));  
fmt::print("{} ", j);    // ~ ><(((°>
```

```
int i(int(x));  
//fmt::print("{} ", i); // doesn't compile
```

```
int j((int(x)));  
fmt::print("{} ", j);    // ~ ~ ><(((°>
```

```
int i(int(x));  
//fmt::print("{} ", i); // doesn't compile
```

```
int j((int(x)));  
fmt::print("{} ", j);    // ~ ~ ~ ~ ><(((°>
```

```
int i(int(x));  
//fmt::print("{} ", i); // doesn't compile
```

```
int j((int(x)));  
fmt::print("{} ", j);    // - ok. ~ ~ ~ ~ >
```

```
int i(int(x));  
//fmt::print("{} ", i); // doesn't compile
```

```
int j((int(x)));  
fmt::print("{} ", j);    // - ok.
```



```
#include <vector>
```

```
int main()
```

```
{
```

```
    std::vector<int> v {{4,3,2,1}};
```

```
    return v.size();
```

```
}
```

```
#include <vector>
```

```
int main()
```

```
{
```

```
    std::vector<int> v {{4,3,2,1}};
```

```
    return v._M_impl._M_end_of_storage - v._M_impl._M_start;
```

```
}
```

```
#define private public
#define protected public
#include <vector>

int main()
{
    std::vector<int> v {{4,3,2,1}};
    return v._M_impl._M_end_of_storage - v._M_impl._M_start;
}
```

```
std::vector<std::uint8_t> a;
```

```
std::vector<std::uint8_t> a;  
a.resize(1 * sizeof a);
```

```
std::vector<std::uint8_t> a;  
a.resize(1 * sizeof a);  
  
a = std::move(*(new (a.data()) std::vector<std::uint8_t>));
```

```
std::vector<std::uint8_t> a;  
a.resize(1 * sizeof a);  
  
a = std::move(*(new (a.data()) std::vector<std::uint8_t>));  
  
// a lives in a!  
  
//travisdowns.github.io/blog/2019/08/26/vector-inc.html
```

```
static_assert(std::is_same_v<signed int, int>); // ok
```



```
static_assert(std::is_same_v<signed int, int>); // ok
static_assert(std::is_same_v<signed int, int signed>); // ok
static_assert(std::is_same_v<signed char, char>);
```

```
static_assert(std::is_same_v<signed int, int>); // ok
static_assert(std::is_same_v<signed int, int signed>); // ok
static_assert(std::is_same_v<signed char, char>); // NOK!!!
```

```
typedef signed int s1;
```

```
typedef signed int s1;
```

```
signed int typedef s3;
```

```
typedef signed int s1;  
signed typedef int s2;  
signed int typedef s3;
```

```
typedef signed int s1;  
signed typedef int s2;  
signed int typedef s3;
```



```
int main()
{
    int x = 100;

    return x;
}
```

```
int main()
{
    int x = 100;
    x = x---x
        |x--|
        x---x;
    return x;
}
```



```
class A
{
public:
    virtual ~A() = 0;

    virtual void foo() = 0;
};
```

```
int main()
{
}
```

```
class A
{
public:
    virtual ~A() = 0;

    virtual void foo() = 0;
};
```

```
struct B : public A
{ void foo() override {} };
```

```
int main()
{ B b; }
```

```
class A
{
public:
    virtual ~A() = 0;

    virtual void foo() = 0;
};

inline A::~~A()
{ fmt::print("purist mode not supported"); }

struct B : public A
{ void foo() override {} };

int main()
{ B b; }
```

```
struct A
{
    static int f() { return 4; }
};

int main()
{

}
```

```
struct A
{
    static int f() { return 4; }
};

int main()
{
    A* a = nullptr;

    return a->f();
}
```

```
struct A
{
    int (&foo() const)[10]
    {
        static int foo[10];
        return foo;
    }
};
```

```
#include <stdio>
```

```
int main()  
{  
    typedef int myint[puts("hi")];  
}
```

```
#include <stdio>
```

```
int main()  
{  
    typedef int myint[puts("hi")];  
}
```





Thank you.