

```
1 class Date {
2     int d, m, y;
3 public:
4     Date(int day, int month, int year) {
5         d = day;
6         m = month;
7         y = year;
8     }
9     int day() {
10        return d;
11    }
12    int month() {
13        return this->m;
14    }
15    int year() {
16        return y;
17    }
18 };
```

Please criticize this code.

foo.hpp

```
1 #include <iostream>
2
3 class Foo {
4     int value;
5     int read_counter;
6 public:
7     Foo(int v) : value(v), read_counter(0) { }
8     int value();
9     void debug_print_read_counter();
10 };
```

foo.cpp

```
1 #include "foo.hpp"
2
3 int Foo::value() const {
4     read_counter++;
5     return value;
6 }
7
8 void Foo::debug_print_read_counter() {
9     std::cerr << "read_counter = "
10                << read_counter << std::endl;
11 }
```

main.cpp

```
1 #include "foo.hpp"
2
3 int main() {
4     Foo f(42);
5     f.debug_print_read_counter();
6     f.value();
7     f.value();
8     f.debug_print_read_counter();
9 }
```

```
g++ foo.cpp main.cpp && ./a.out
```

This code does not compile and it smells bad. How can you fix it? Please criticize.

```

1 #include <iostream>
2
3 template<typename T> void p(T x) { std::cout << x; }
4
5 struct B {
6     B() { p('B'); }
7     ~B() { p('b'); }
8 };
9
10 class A {
11     B * v;
12 public:
13     A() { p('A'); v = new B[3]; }
14     ~A() { p('a'); delete v; }
15 };
16
17 int main() {
18     A a1;
19     A a2 = a1;
20     A a3;
21     a3 = a1;
22 }

```

What might happen if you try to compile, link and run this program?
Please criticize.

```

1 #include <iostream>
2
3 template<typename T> void p(T x) { std::cout << x; }
4
5 struct A {
6     A() { p('A'); }
7     ~A() { p('a'); }
8 };
9
10 struct B {
11     B() { p('B'); }
12     ~B() { p('b'); }
13 };
14
15 class C {
16     A a;
17     B b;
18 public:
19     C() : b(), a() {}
20     C(int) {}
21 };
22
23 int main() {
24     C c1;
25     C c2(4);
26 }

```

what might happen if you try to compile, link and run this program?

```

1 #include <iostream>
2
3 template<typename T> void p(T x) { std::cout << x; }
4
5 struct A {
6     int value;
7     A() { p(0); }
8     A(int v) : value(v) { p(1); }
9     ~A() { p(2); }
10    A(const A & a) { p(3); }
11    void operator=(const A & a) { p(4); }
12 };
13
14 struct B {
15     A a;
16     B(int v) { p('B'); a=42; p('b'); }
17 };
18
19 struct C {
20     A a;
21     C(int v) : a(v) { p('C'); p('c'); }
22 };
23
24 int main() {
25     p('-');
26     B b(42);
27     p('-');
28     C c(42);
29     p('-');
30 }

```

What might happen if you try to compile, link and run this program?

```
1 #include <iostream>
2
3 template<typename T> void p(T x) { std::cout << x; }
4
5 struct X {
6     int a;
7     int b;
8     X() : a(1), b(2) {}
9 };
10
11 struct Y {
12     int c;
13     int d;
14     Y() : c(3), d(4) {}
15 };
16
17 int main() {
18     X x;
19     new (&x) Y;
20     p(x.a);
21     p(x.b);
22 }
```

what might happen if you try to compile, link and run this program?

```

1 #include <iostream>
2
3 template<typename T> void p(T x) { std::cout << x; }
4
5 struct X {
6     int value;
7     X(int v) : value(v) { p('X'); p(v); }
8     ~X() { p('x'); }
9     X(const X & x) : value(x.value) { p(2); }
10    X operator+(const X & x) { p(3); return this->value + x.value; }
11 };
12
13 int main() {
14     X a(4);
15     p('-');
16     p( (a + a).value );
17     p('-');
18     X b = (a + a);
19     p('-');
20     p(b.value);
21     p('-');
22 }

```

what might happen if you try to compile, link and run this program?

```
1 #include <iostream>
2
3 template<typename T> void p(T x) { std::cout << x; }
4
5 struct X {
6     X(int v) { p('X'); p(v); }
7     ~X() { p('x'); }
8 };
9
10 void foo(X x) {
11     p('F');
12 }
13
14 int main() {
15     foo(4);
16 }
```

what might happen if you try to compile, link and run this program?


```
1 #include <iostream>
2
3 template<typename T> void p(T x) { std::cout << x; }
4
5 struct X {
6     int value;
7     X(int v) : value(v) { }
8     operator int() const { return value; }
9 };
10
11 void foo(int v) {
12     p(v);
13 }
14
15 int main() {
16     X x = 42;
17     foo(x);
18 }
```

what might happen if you try to compile, link and run this program?

```

1 #include <iostream>
2 #include <vector>
3
4 class X {
5     int value;
6 public:
7     X(int v) : value(v) {}
8     void operator()(int & i) const {
9         i += value;
10    }
11 };
12
13 void p(int & i) {
14     std::cout << i << std::endl;
15 }
16
17 int main() {
18     std::vector<int> v;
19     v.push_back(1);
20     v.push_back(2);
21     v.push_back(3);
22     for_each(v.begin(), v.end(), X(42));
23     for_each(v.begin(), v.end(), p);
24 }

```

what might happen if you try to compile, link and run this program?