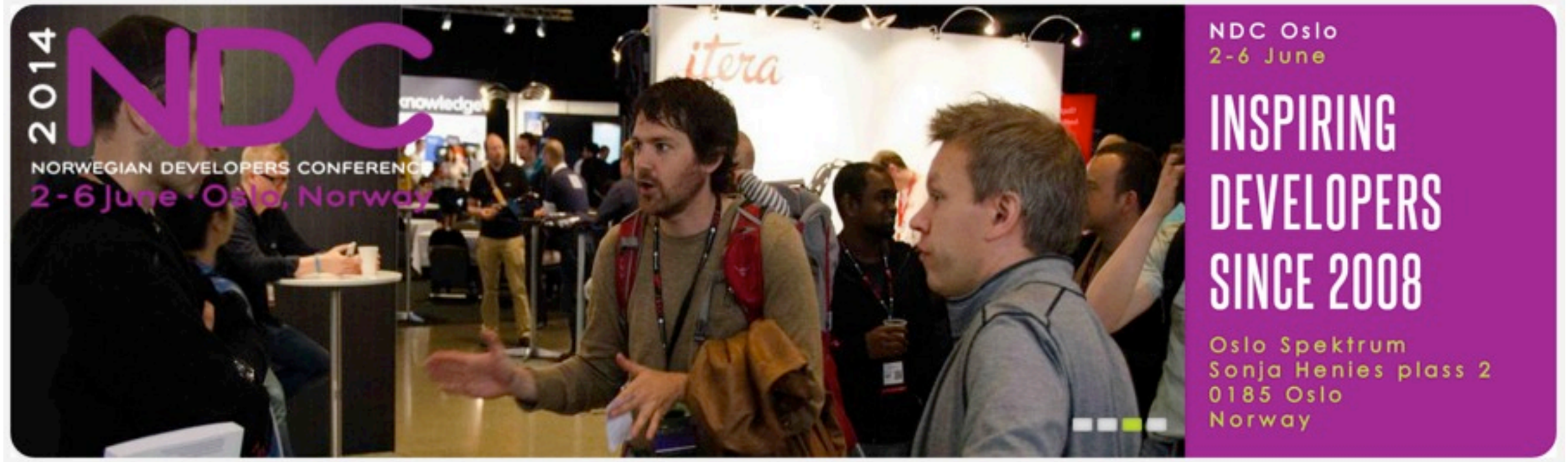
C++ Pub Quiz

Olve Maudal, feat Lars Gullik

++

A 90 minute quiz session at ACCU
Terrace Bar, Marriott Hotel, Bristol
1600-1730, April 10, 2014

Sponsored by NDC conferences:



The question for all code snippets is:

*What will actually happen on **my** machine?*

Full score is given if you manage to guess:

*Whatever actually happens on **my** machine!*

There are few trick questions here, most/all of the code snippets do produce the expected result and should be quite easy if you really understand 🍺++

PS: All the code snipppets do indeed compile, link and run on ***my*** machine. There are no missing semicolons or syntax errors in the snippets. The output is always a straightforward sequence of non-whitespace characters.

Disclaimer: the code snippets here are all crap examples of how to write code. This is just for fun.

Remember, this is **not** about C++, nor G++, it is about:

++

Sponsored by:

"My machine" is a plain, up-to-date 32-bit Ubuntu 13.10 with g++ 4.8.1

```
$ uname -a
Linux oma-VirtualBox 3.11.0-12-generic #19-Ubuntu SMP Wed Oct 9 16:12:00 UTC 2013 i686
i686 i686 GNU/Linux
$ g++ --version
g++ (Ubuntu/Linaro 4.8.1-10ubuntu9) 4.8.1
$ cat /proc/cpuinfo
processor       : 0
vendor_id       : GenuineIntel
cpu family      : 6
model           : 58
model name      : Intel(R) Core(TM) i7-3635QM CPU @ 2.40GHz
stepping        : 9
microcode       : 0x19
cpu MHz         : 2410.104
cache size      : 6144 KB
fdiv_bug        : no
f00f_bug        : no
coma_bug        : no
fpu             : yes
fpu_exception   : yes
cpuid level     : 5
wp              : yes
flags           : fpu vme de pse tsc msr pae mce cx8 apic sep mtrr pge mca cmov pat pse36
clflush mmx fxsr sse sse2 nx rdtscp constant_tsc pni monitor ssse3
bogomips        : 4820.20
clflush size    : 64
cache_alignment : 64
address sizes   : 36 bits physical, 48 bits virtual
power management:

$ g++ -std=c++11 -Wall -Wextra -O -pthread foo.cpp
$ ./a.out
```

(Nothing to see here, move along please...)

Many of the questions are very much inspired by stuff that was submitted to cppquiz.org developed and maintained by Anders Knatten

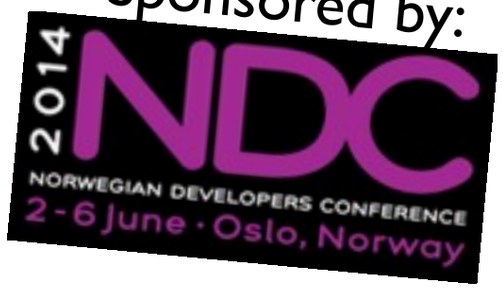
Other major sources of inspiration: Jonathan, Nico, Andrew, Lars Gullik and stuff that I stumble over on the internet.

Thanks!

++ C++ Pub Quiz

Olve Maudal, feat Lars Gullik

Sponsored by:



Quiz	Answer	Notes	Score	Bonus
0				
1				
2				
3				
4				
5				
6				
7				
8				
9				
10				

Team name:

The Destructors

Start bonus	Score	Bonus	Total
10			

10 points as start bonus
3 points for each correct answer
1 point for no answer
-1 point for an incorrect answer

For some of the answers there are bonus points.

Questions

#0

```
#include <iostream>

int main() {
    int a[]{1,2,3,4};
    std::cout << 0;
    for (auto x : a)
        std::cout << x;
    std::cout << 9;
}
```

++ C++ Pub Quiz

Olve Maudal, feat Lars Gullik

Quiz	Answer	Notes	Score	Bonus
0	0 1 2 3 4 9	auto ref, C++17 for(x:a)		
1				
2				
3				
4				
5				
6				
7				
8				
9				
10				

Team name:

The Destructors

Start bonus	Score	Bonus	Total
10			

#I

```
#include <iostream>

void foo(double) { std::cout << "a"; }
void foo(float) { std::cout << "b"; }
void foo(const std::string &) { std::cout << "c"; }
void foo(const void *) { std::cout << "d"; }
void foo(int) { std::cout << "e"; }
void foo(unsigned int) { std::cout << "f"; }
void foo(short) { std::cout << "j"; }
void foo(long) { std::cout << "k"; }

int main() {
    foo(1.0);
    foo("hello");
    foo(0x80000000);
    foo(0x7FFFFFFF);
    foo(NULL);
    short s = 3;
    foo(s);
    foo(s + s);
    long l = 3;
    foo(l);
    foo(l + l);
}
```

#2

```
#include <iostream>
#include <cctype>

struct X {
    X(char ch) : ch_(ch) { std::cout << ch_; }
    ~X() { std::cout << (char)std::toupper(ch_); }
    char ch_;
};

X a('a');

int main()
{
    int counter = 0;
    X b('b');
  label1:
    X c('c');
    if (counter++ == 0)
        goto label1;
    static X d('d');
    X('e');
    X f('f');
    do {
        X g('g');
        if (counter++ == 1)
            continue;
    } while (false);
}

static X h('h');
```

#3

```
#include <iostream>
#include <utility>

void y(int&) { std::cout << '1'; }
void y(int&&) { std::cout << '2'; }

template <typename T> void f(T && x) { y(x); }
template <typename T> void g(T && x) { y(std::move(x)); }
template <typename T> void h(T && x) { y(std::forward<T>(x)); }

int main() {
    int i = 42;
    y(i), y(42);
    f(i), f(42);
    g(i), g(42);
    h(i), h(42);
}
```

#4

```
#include <iostream>

struct A {
    A() { std::cout << 0; foo(); }
    ~A() { std::cout << 9; foo(); }
    virtual void foo() const { std::cout << 1; }
    virtual void bar() const { std::cout << 2; foo(); }
};

struct B : public A {
    virtual void foo() const { std::cout << 3; }
    void bar() const { std::cout << 4; foo(); }
};

struct C : public A {
    void foo() const { std::cout << 5; }
    virtual void bar() { std::cout << 6; foo(); }
};

int main() {
    const A & a = B();
    a.bar();
    static C c;
    c.bar();
}
```

#5

```
#include <iostream>

struct S {
    short a;
    char b;
};

class C {
    short a;
    char b;
};

class EC : C {
    char c;
};

struct ES : S {
    char c;
};

int main() {
    static_assert(sizeof(short) == 2,"");
    static_assert(sizeof(S) == 4,"");
    std::cout << sizeof(C);
    std::cout << sizeof(EC);
    std::cout << sizeof(ES);
    return 0;
}
```

```
#include <iostream>

template <template <typename> class>
struct X {
  X() { std::cout << "1"; }
};

template <typename>
struct Y {};

template <typename T>
using Z = Y<T>;

template <>
struct X<Y> {
  X() { std::cout << "2"; }
};

int main() {
  X<Y> x1;
  X<Z> x2;
}
```

#7

```
#include <iostream>

struct P {
    P() { std::cout << '1'; };
    P(int) { std::cout << '2'; };
    P(int, int) { std::cout << '3'; };
    P(std::initializer_list<int>) { std::cout << '4'; }
};

int main()
{
    P a();
    P b(11);
    P c(11,12);

    P e{};
    P f{11};
    P g{11,12};

    P h = {};
    P i = {11};
    P j = {11,12};

    P k = (1,2);
    P l = {{1,2}};
    P m = (1,2,3);
    P n = {1,2,3};
}
```

#8

```
#include <iostream>
#include <string>
#include <future>

void foo() { std::cout << 1 << std::flush; }
void bar() { std::cout << 2 << std::flush; }
void gaz() { std::cout << 3 << std::flush; }

int main() {

    std::async(foo);
    std::async(std::launch::async, bar);
    std::async(std::launch::deferred, gaz);

    std::this_thread::sleep_for(std::chrono::milliseconds(1000));

    std::cout << 9 << std::flush;


    auto f1 = std::async(foo);
    auto f2 = std::async(std::launch::async, bar);
    auto f3 = std::async(std::launch::deferred, gaz);

    std::this_thread::sleep_for(std::chrono::milliseconds(1000));

    f1.get();
    f2.get();
    f3.get();
}
```

```
#include <iostream>

struct X {
    X(const char *) { std::cout << 1; }
    X(const X &) { std::cout << 2; }
    X(X &&) { std::cout << 3; }
};

X f(X a) {
    return a;
}

X g(const char * b) {
    X c(b);
    return c;
}

int main() {
    f("hello");
    g("hello");
}
```

#10

```
#include <iostream>

template<typename T> void P(T x) { std::cout << x; }

void foo(char a) {
    P(3);
    P(a);
}

template <typename... A>
void foo(int a, A... args) {
    foo(args...);
    P(a);
}

template <typename... A>
void foo(char a, A... args) {
    P(a);
    foo(args...);
}

int main()
{
    foo('1','2',48,'4','5');
}
```

Sponsored by:

Answers

#0

```
#include <iostream>

int main() {
    int a[]{1,2,3,4};
    std::cout << 0;
    for (auto x : a)
        std::cout << x;
    std::cout << 9;
}
```

```
012349
```

1 bonus point "auto &" was discussed

1 bonus point if the C++17 proposal
"for (x : a)" was discussed

++ C++ Pub Quiz

Olve Maudal, feat Lars Gullik

Quiz	Answer	Notes	Score	Bonus
0	0 1 2 3 4 9	auto ref, C++17 for(x:a)	3	2
1				
2				
3				
4				
5				
6				
7				
8				
9				
10				

Team name:

The Destructors

Start bonus	Score	Bonus	Total
10			

#1

```
#include <iostream>

void foo(double) { std::cout << "a"; }
void foo(float) { std::cout << "b"; }
void foo(const std::string &) { std::cout << "c"; }
void foo(const void *) { std::cout << "d"; }
void foo(int) { std::cout << "e"; }
void foo(unsigned int) { std::cout << "f"; }
void foo(short) { std::cout << "j"; }
void foo(long) { std::cout << "k"; }

int main() {
    foo(1.0);
    foo("hello");
    foo(0x80000000);
    foo(0x7FFFFFFF);
    foo(NULL);
    short s = 3;
    foo(s);
    foo(s + s);
    long l = 3;
    foo(l);
    foo(l + l);
}
```

```
adfeejekk
```

1 bonus point if "nullptr" was discussed

2 consolation points if you got adfekjekk

++ C++ Pub Quiz

Olve Maudal, feat Lars Gullik

Quiz	Answer	Notes	Score	Bonus
0	0 1 2 3 4 9	auto ref, C++17 for(x:a)	3	2
1	b d f e e j e k k	nullptr	-1	1
2				
3				
4				
5				
6				
7				
8				
9				
10				

Team name:

The Destructors

Start bonus	Score	Bonus	Total
10			

#2

```
#include <iostream>
#include <cctype>

struct X {
    X(char ch) : ch_(ch) { std::cout << ch_; }
    ~X() { std::cout << (char)std::toupper(ch_); }
    char ch_;
};

X a('a');

int main()
{
    int counter = 0;
    X b('b');
  label1:
    X c('c');
    if (counter++ == 0)
        goto label1;
    static X d('d');
    X('e');
    X f('f');
    do {
        X g('g');
        if (counter++ == 1)
            continue;
    } while (false);
}

static X h('h');
```

ahbcCcdeEfgGFCBDHA

1 bonus point if "don't use goto" was discussed

1 bonus point if "don't use continue" was discussed

1 bonus point if "send Olve back to Norway" or something similar was discussed

#3

```
#include <iostream>
#include <utility>

void y(int&) { std::cout << '1'; }
void y(int&&) { std::cout << '2'; }

template <typename T> void f(T && x) { y(x); }
template <typename T> void g(T && x) { y(std::move(x)); }
template <typename T> void h(T && x) { y(std::forward<T>(x)); }

int main() {
    int i = 42;
    y(i), y(42);
    f(i), f(42);
    g(i), g(42);
    h(i), h(42);
}
```

```
12112212
```

1 bonus point if "universal reference" was discussed

2 bonus point if "reference collapsing" was discussed

#4

```
#include <iostream>

struct A {
    A() { std::cout << 0; foo(); }
    ~A() { std::cout << 9; foo(); }
    virtual void foo() const { std::cout << 1; }
    virtual void bar() const { std::cout << 2; foo(); }
};

struct B : public A {
    virtual void foo() const { std::cout << 3; }
    void bar() const { std::cout << 4; foo(); }
};

struct C : public A {
    void foo() const { std::cout << 5; }
    virtual void bar() { std::cout << 6; foo(); }
};

int main() {
    const A & a = B();
    a.bar();
    static C c;
    c.bar();
}
```

01430165919 1

I bonus point if "the override specifier" was discussed

#5

```
#include <iostream>

struct S {
    short a;
    char b;
};

class C {
    short a;
    char b;
};

class EC : C {
    char c;
};

struct ES : S {
    char c;
};

int main() {
    static_assert(sizeof(short) == 2,"");
    static_assert(sizeof(S) == 4,"");
    std::cout << sizeof(C);
    std::cout << sizeof(EC);
    std::cout << sizeof(ES);
    return 0;
}
```

4 4 6

1 bonus point if "alignment" was discussed

#6

```
#include <iostream>

template <template <typename> class>
struct X {
  X() { std::cout << "1"; }
};

template <typename>
struct Y {};

template <typename T>
using Z = Y<T>;

template <>
struct X<Y> {
  X() { std::cout << "2"; }
};

int main() {
  X<Y> x1;
  X<Z> x2;
}
```

21

1 bonus point if WTF was said at least two times during the discussion

#7

```
#include <iostream>

struct P {
    P() { std::cout << '1'; };
    P(int) { std::cout << '2'; };
    P(int, int) { std::cout << '3'; };
    P(std::initializer_list<int>) { std::cout << '4'; }
};

int main()
{
    P a();
    P b(11);
    P c(11,12);

    P e{};
    P f{11};
    P g{11,12};

    P h = {};
    P i = {11};
    P j = {11,12};

    P k = (1,2);
    P l = {{1,2}};
    P m = (1,2,3);
    P n = {1,2,3};
}
```

2 3 1 4 4 1 4 4 2 4 2 4

1 bonus point if "explicit constructor" was mentioned in your discussion.

1 bonus point if Nico Josuttis' ACCU2014 talk was mentioned in your discussion.

1 consolation point if you got 1 2 3 1 4 4 1 4 4 2 4 2 4

#8

```
#include <iostream>
#include <string>
#include <future>

void foo() { std::cout << 1 << std::flush; }
void bar() { std::cout << 2 << std::flush; }
void gaz() { std::cout << 3 << std::flush; }

int main() {

    std::async(foo);
    std::async(std::launch::async, bar);
    std::async(std::launch::deferred, gaz);

    std::this_thread::sleep_for(std::chrono::milliseconds(1000));

    std::cout << 9 << std::flush;


    auto f1 = std::async(foo);
    auto f2 = std::async(std::launch::async, bar);
    auto f3 = std::async(std::launch::deferred, gaz);

    std::this_thread::sleep_for(std::chrono::milliseconds(1000));

    f1.get();
    f2.get();
    f3.get();
}
```

```
29213
```

1 bonus point if "deprecate async" was discussed

#9

```
#include <iostream>

struct X {
    X(const char *) { std::cout << 1; }
    X(const X &) { std::cout << 2; }
    X(X &&) { std::cout << 3; }
};

X f(X a) {
    return a;
}

X g(const char * b) {
    X c(b);
    return c;
}

int main() {
    f("hello");
    g("hello");
}
```

131

1 bonus point if "return value optimization" was discussed

1 bonus point if Jonathan Wakely and accu-general was discussed

#10

```
#include <iostream>

template<typename T> void P(T x) { std::cout << x; }

void foo(char a) {
    P(3);
    P(a);
}

template <typename... A>
void foo(int a, A... args) {
    foo(args...);
    P(a);
}

template <typename... A>
void foo(char a, A... args) {
    P(a);
    foo(args...);
}

int main()
{
    foo('1','2',48,'4','5');
}
```

```
12355248
```

3 bonus point if the PubQuiz at ACCU 2012 was mentioned

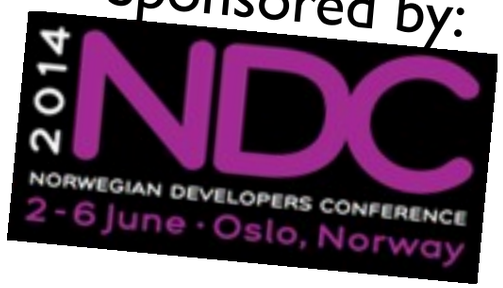
10 points as start bonus
3 points for each correct answer
1 point for no answer
-1 point for an incorrect answer

For some of the answers there are bonus points.

++ C++ Pub Quiz

Olve Maudal, feat Lars Gullik

Sponsored by:



Quiz	Answer	Notes	Score	Bonus
0	0 1 2 3 4 9	auto ref, C++17 for(x:a)	3	2
1	a d f e e j e k k	nullptr	3	1
2	ahbcCcdeEfgGFCBDHA	goto?, continue?	3	3
3	12 11 22 12	universal reference, reference collapsing	3	3
4	0 1 4 3 0 1 6 5 9 1 9 1	override specifier	3	1
5	4 4 6	alignment?	3	1
6	2 1	wtf, wtf, wtf	3	1
7	2 3 1 4 4 1 4 4 2 4 2 4	explicit ctor, Nico	3	2
8	2 9 2 1 3	was async a mistake?	3	1
9	1 3 1	rvo, jonathan posted this on accu-general	3	2
10	1 2 3 5 5 2 4 8	ACCU 2012?	3	3

Team name:

The Destructors

Start bonus	Score	Bonus	Total
10	33	20	63

The winners of ACCU 2014 C++ Pub Quiz (9 teams)

`Evil vector<bool>`

With an amazing score of 41
(the runners up had 32, 29,29,29)

Sponsored by: