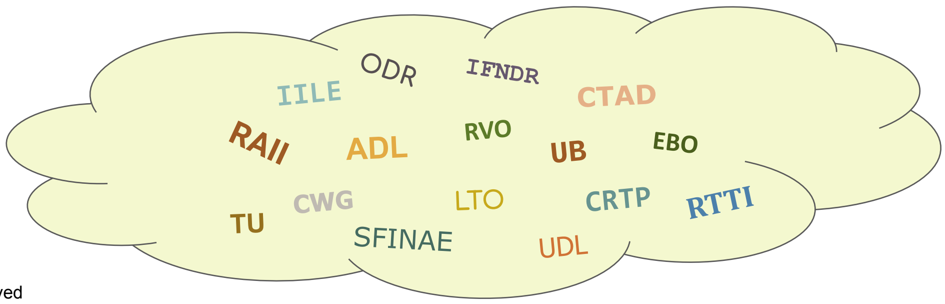
CppCon
The C++ Conference
2021
October 24-29
IN AURORA COLORADO USA AND ONLINE

Understanding and Mastering C++'s Complexity

Amir Kirsh



Before we `.begin()`

Before we `.begin()`

Does the code below compile? If so, how? If not, why?

(Wait… question is open only for C++ programmers with 0-3 years of experience)

Before we `.begin()`

Does the code below compile? If so, how? If not, why?

```cpp
#include <utility>
#include <string>

int main() {
    std::string s1 = "hi", s2 = "bye";
    swap(s1, s2);

    int a = 3, b = 7;
    swap(a, b);
}
```

Complexity

What this talk is about (and what's not)

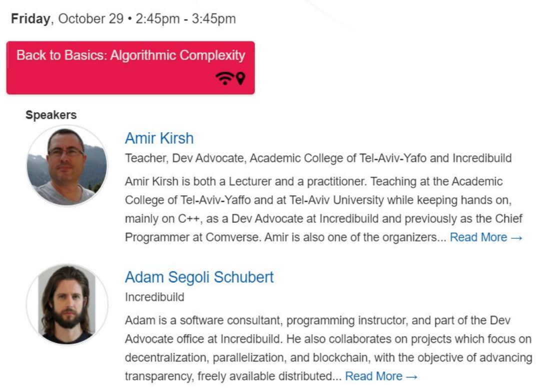
Complexity

This talk is **NOT** about Algorithmic Complexity (no big 'O' in this talk!)

Complexity

This talk is **NOT** about Algorithmic Complexity (no big 'O' in this talk!)

But we do have a talk on
Algorithmic Complexity - on Friday!



Friday, October 29 • 2:45pm - 3:45pm

Back to Basics: Algorithmic Complexity

Speakers

Amir Kirsh
Teacher, Dev Advocate, Academic College of Tel-Aviv-Yafo and Incredibuild
Amir Kirsh is both a Lecturer and a practitioner. Teaching at the Academic College of Tel-Aviv-Yaffo and at Tel-Aviv University while keeping hands on, mainly on C++, as a Dev Advocate at Incredibuild and previously as the Chief Programmer at Comverse. Amir is also one of the organizers... Read More →

Adam Segoli Schubert
Incredibuild
Adam is a software consultant, programming instructor, and part of the Dev Advocate office at Incredibuild. He also collaborates on projects which focus on decentralization, parallelization, and blockchain, with the objective of advancing transparency, freely available distributed... Read More →

Complexity

This talk is about **C++ language complexity**,
with a broad definition for complexity:

anything that ***makes it hard for you to use C++***, or ***to understand it***,
including things that ***irritate or annoy*** you, things that ***waste your time***,
and language syntax that is ***bug prone***, or ***broken*** in a way, or is ***done easier*** in other languages.

About Me

Amir Kirsh

Lecturer
Academic College of Tel-Aviv-Yaffo
and Tel-Aviv University

Developer Advocate
INCREDIBUILD

Co-Organizer of the **CoreCpp** conference and meetup group

Talk Origins

A graduated student of mine was interviewed for a C++ position and consulted me whether C++ is a right choice (as “there are other less complex languages”).

Talk Origins

A graduated student of mine was interviewed for a C++ position and consulted me whether C++ is a right choice (as “there are other less complex languages”).

I convinced her.

Talk Origins

A graduated student of mine was interviewed for a C++ position and consulted me whether C++ is a right choice (as “there are other less complex languages”).

I convinced her.

She is now a C++ developer at Waves.com

Amit Barzilay

Talk Origins

A graduated student of mine was interviewed for a C++ position and consulted me whether C++ is a right choice (as “there are other less complex languages”).

I convinced her.

She is now a C++ developer at Waves.com

Amit Barzilay

We prepared this talk together for CoreCpp conference 2021 in Tel-Aviv.

And now you got me here :-)

Complexity

Isn't it the name of the game? (of being a programmer...)

The Perils of Java Schools

Joel Spolsky, 2005

Lazy kids.
Whatever happened to hard work?
...
in the last decade a large number of otherwise perfectly good schools have gone 100% Java ...
The lucky kids of JavaSchools are never going to get weird segfaults trying to implement pointer-based hash tables. They're never going to go stark, raving mad trying to pack things into bits.

https://www.joelonsoftware.com/2005/12/29/the-perils-of-javaschools-2

Complexity

Is C++ complex?

Why?

Can it be less complex?

What makes C++ complex - *for you*?

```
Please check only the things that bother
you in person, don’t check topics that you
are not familiar with or don’t use at all:
```

What makes C++ complex - *for you*?

Please check only the things that bother you in person, don’t check topics that you are not familiar with or don’t use at all:

What makes C++ complex - *for you*?

Let’s review the questionnaire results

Complexity

Complexity

Why programming is complex?

What makes a software language complex?

What makes a software language complex?

- Too many ways for doing the same thing (“too many options”)

What makes a software language complex?

- Too many ways for doing the same thing (“too many options”)
- Too few ways for doing things (“hard to express yourself fluently”)

What makes a software language complex?

- Too many ways for doing the same thing (“too many options”)
- Too few ways for doing things (“hard to express yourself fluently”)
- Lack of standard / documentation / proper examples / community

What makes a software language complex?

- Too many ways for doing the same thing (“too many options”)
- Too few ways for doing things (“hard to express yourself fluently”)
- Lack of standard / documentation / proper examples / community
- Complex model
  - Too low level
  - Too high-level, abstract
  - Contradicting paradigms or rules
  - Rules are not intuitive or too complicated

What makes a software language complex?

- Too many ways for doing the same thing (“too many options”)
- Too few ways for doing things (“hard to express yourself fluently”)
- Lack of standard / documentation / proper examples / community
- Complex model
  - Too low level
  - Too high-level, abstract
  - Contradicting paradigms or rules
  - Rules are not intuitive or too complicated
- Complex problems

What makes a software language complex?

- Too many ways for doing the same thing ("too many options")
- Too few ways for doing things ("hard to express yourself fluently")
- Lack of standard / documentation / proper examples / community
- Complex model
  - Too low level
  - Too high-level, abstract
  - Contradicting paradigms or rules
  - Rules are not intuitive or too complicated
- Complex problems
- New syntax, new stuff getting into the language

What makes a software language complex?

- Too many ways for doing the same thing ("too many options")
- Too few ways for doing things ("hard to express yourself fluently")
- Lack of standard / documentation / proper examples / community
- Complex model
  - Too low level
  - Too high-level, abstract
  - Contradicting paradigms or rules
  - Rules are not intuitive or too complicated
- Complex problems
- New syntax, new stuff getting into the language

- Backward compatibility issues between language versions

What makes a software language complex?

- Too many ways for doing the same thing ("too many options")
- Too few ways for doing things ("hard to express yourself fluently")
- Lack of standard / documentation / proper examples / community
- Complex model
  - Too low level
  - Too high-level, abstract
  - Contradicting paradigms or rules
  - Rules are not intuitive or too complicated
- Complex problems
- New syntax, new stuff getting into the language
- Backward compatibility issues between language versions
- Lack of proper tools

What makes a software language complex?

- Too many ways for doing the same thing ("too many options")
- Too few ways for doing things ("hard to express yourself fluently")
- Lack of standard / documentation / proper examples / community
- Complex model
  - Too low level
  - Too high-level, abstract
  - Contradicting paradigms or rules
  - Rules are not intuitive or too complicated
- Complex problems
- New syntax, new stuff getting into the language
- Backward compatibility issues between language versions
- Lack of proper tools

Complexity

Is C++ complex?

Why?

Can it be less complex?

What makes C++ complex?

- Too many ways for doing the same thing ("too many options")
- Too few ways for doing things ("hard to express yourself fluently")
- Lack of standard / documentation / proper examples / community
- Complex model
  - Too low level
  - Too high-level, abstract
  - Contradicting paradigms or rules
  - Rules are not intuitive or too complicated
- Complex problems
- New syntax, new stuff getting into the language
- Backward compatibility issues between language versions
- Lack of proper tools

What makes C++ complex?

- Too many ways for doing the same thing ("too many options")?
- Too few ways for doing things ("hard to express yourself fluently")?

What makes C++ complex?

- ~~Too many ways for doing the same thing ("too many options")~~
- ~~Too few ways for doing things ("hard to express yourself fluently")~~

What makes C++ complex?

- ~~Too many ways for doing the same thing (“too many options”)~~
- ~~Too few ways for doing things (“hard to express yourself fluently”)~~
- Lack of standard / documentation / proper examples / community?

What makes C++ complex?

- ~~Too many ways for doing the same thing (“too many options”)~~
- ~~Too few ways for doing things (“hard to express yourself fluently”)~~
- ~~Lack of standard / documentation / proper examples / community~~

What makes C++ complex?

- ~~Too many ways for doing the same thing (“too many options”)~~
- ~~Too few ways for doing things (“hard to express yourself fluently”)~~
- ~~Lack of standard / documentation / proper examples / community~~
- Complex model
    - Too low level
    - Too high-level, abstract
    - Contradicting paradigms or rules
    - Rules are not intuitive or too complicated

What makes C++ complex?

- ~~Too many ways for doing the same thing (“too many options”)~~
- ~~Too few ways for doing things (“hard to express yourself fluently”)~~
- ~~Lack of standard / documentation / proper examples / community~~
- Complex model
    - ~~Too~~ Has quite a few low level features
    - ~~Too~~ Has quite a few high-level features
    - Has some contradicting paradigms or rules
    - Some rules are not intuitive ~~or too complicated~~

What makes C++ complex?

- ~~Too many ways for doing the same thing (“too many options”)~~
- ~~Too few ways for doing things (“hard to express yourself fluently”)~~
- ~~Lack of standard / documentation / proper examples / community~~
- Complex model
  - ~~Too~~ Has quite a few low level features
  - ~~Too~~ Has quite a few high-level features
  - Has some contradicting paradigms or rules
  - Some rules are not intuitive ~~or too complicated~~
- Complex problems?

What makes C++ complex?

- ~~Too many ways for doing the same thing (“too many options”)~~
- ~~Too few ways for doing things (“hard to express yourself fluently”)~~
- ~~Lack of standard / documentation / proper examples / community~~
- Complex model
  - ~~Too~~ Has quite a few low level features
  - ~~Too~~ Has quite a few high-level features
  - Has some contradicting paradigms or rules
  - Some rules are not intuitive ~~or too complicated~~
- Deals frequently with complex problems

What makes C++ complex?

- ~~Too many ways for doing the same thing ("too many options")~~
- ~~Too few ways for doing things ("hard to express yourself fluently")~~
- ~~Lack of standard / documentation / proper examples / community~~
- Complex model
  - ~~Too~~ Has quite a few low level features
  - ~~Too~~ Has quite a few high-level features
  - Has some contradicting paradigms or rules
  - Some rules are not intuitive ~~or too complicated~~
- Deals frequently with complex problems
- New syntax, new stuff getting into the language?

What makes C++ complex?

- ~~Too many ways for doing the same thing ("too many options")~~
- ~~Too few ways for doing things ("hard to express yourself fluently")~~
- ~~Lack of standard / documentation / proper examples / community~~
- Complex model
  - ~~Too~~ Has quite a few low level features
  - ~~Too~~ Has quite a few high-level features
  - Has some contradicting paradigms or rules
  - Some rules are not intuitive ~~or too complicated~~
- Deals frequently with complex problems
- ~~New syntax, new stuff getting into the language~~

What makes C++ complex?

- ~~Too many ways for doing the same thing ("too many options")~~
- ~~Too few ways for doing things ("hard to express yourself fluently")~~
- ~~Lack of standard / documentation / proper examples / community~~
- Complex model
  - ~~Too~~ Has quite a few low level features
  - ~~Too~~ Has quite a few high-level features
  - Has some contradicting paradigms or rules
  - Some rules are not intuitive ~~or too complicated~~
- Deals frequently with complex problems
- ~~New syntax, new stuff getting into the language~~
- Backward compatibility issues between language versions?

What makes C++ complex?

- ~~Too many ways for doing the same thing ("too many options")~~
- ~~Too few ways for doing things ("hard to express yourself fluently")~~
- ~~Lack of standard / documentation / proper examples / community~~
- Complex model
  - ~~Too~~ Has quite a few low level features
  - ~~Too~~ Has quite a few high-level features
  - Has some contradicting paradigms or rules
  - Some rules are not intuitive ~~or too complicated~~
- Deals frequently with complex problems
- ~~New syntax, new stuff getting into the language~~
- ~~Backward compatibility issues between language versions~~

What makes C++ complex?

- ~~Too many ways for doing the same thing (“too many options”)~~
- ~~Too few ways for doing things (“hard to express yourself fluently”)~~
- ~~Lack of standard / documentation / proper examples / community~~
- Complex model
  - ~~Too~~ Has quite a few low level features
  - ~~Too~~ Has quite a few high-level features
  - Has some contradicting paradigms or rules
  - Some rules are not intuitive ~~or too complicated~~
- Deals frequently with complex problems
- ~~New syntax, new stuff getting into the language~~
- ~~Backward compatibility issues between language versions~~
- Lack of proper tools?

What makes C++ complex?

- ~~Too many ways for doing the same thing (“too many options”)~~
- ~~Too few ways for doing things (“hard to express yourself fluently”)~~
- ~~Lack of standard / documentation / proper examples / community~~
- Complex model
  - ~~Too~~ Has quite a few low level features
  - ~~Too~~ Has quite a few high-level features
  - Has some contradicting paradigms or rules
  - Some rules are not intuitive ~~or too complicated~~
- Deals frequently with complex problems
- ~~New syntax, new stuff getting into the language~~
- ~~Backward compatibility issues between language versions~~
- Lack of proper tools. But improving!

What makes C++ complex?

- Complex model
  - ~~Too~~ Has quite a few low level features
  - ~~Too~~ Has quite a few high-level features
  - Has some contradicting paradigms or rules
  - Some rules are not intuitive ~~or too complicated~~
- Deals frequently with complex problems

What makes C++ complex?

- Complex model
  - ~~Too~~ Has quite a few low level features
  - ~~Too~~ Has quite a few high-level features
  - Has some contradicting paradigms or rules
  - Some rules are not intuitive ~~or too complicated~~

What makes C++ complex?

- Complex model
    - Has some contradicting paradigms or rules
    - Some rules are not intuitive ~~or too complicated~~

What makes C++ complex?

and that's what you hold against a language with >5M users and billions lines of code??

- Complex model
    - Has some contradicting paradigms or rules
    - Some rules are not intuitive ~~or too complicated~~

There is no silver bullet

Picture: https://www.infoq.com/articles/No-Silver-Bullet-Summary -- OOPSLA 2005, Montreal

“No Silver Bullet - Essence and Accident in Software Engineering” by Fred Brooks, 1986

Essence or Accident?

Essence or Accident?

Let’s play...

Essence or Accident?

```
std::map<std::string, std::list<std::pair<Date, Price>>>::iterator quotesItr
     = stocks.find(id);
```

Essence or Accident?

```
std::map<std::string, std::list<std::pair<Date, Price>>>::iterator quotesItr
     = stocks.find(id);
```

C++11

```
auto quotesItr = stocks.find(id);
```

Essence or Accident?

Essence or Accident?

```
What’s the problem here:
```

```cpp
class Shape {
    Color color;
public:
    virtual void draw() const = 0;
    virtual void move(int diffX, int diffY) = 0;
};
```

Can the compiler *deduce* that a class *needs* a virtual destructor and provide one?

Can the compiler *deduce* that a class *needs* a virtual destructor and provide one?

```
What are the actual rules for “you must have a virtual destructor”?
```

Can the compiler *deduce* that a class *needs* a virtual destructor and provide one?

```
What are the actual rules for “you must have a virtual destructor”?
```

```cpp
class Shape {
    Color color;
public:
    virtual void draw() const = 0;
    virtual void move(int diffX, int diffY) = 0;
};
```

```cpp
int main() {
  Rect r ({10, 10}, {20, 20});
  Shape* p = &r;
  p->draw();
}
```

Essence or Accident?

Essence or Accident?

```
template<class K, class V, size_t SIZE, class FetchFunc>
class Cache {
  struct Holder {
    V val;
    mutable typename list<K>::iterator posInList;
    Holder(V v, Date exp, typename list<K>::iterator pos)
    : val(v), posInList(pos), expiry(exp), accessed(Date()) {}
    // ...
  private:
    Date expiry;
    mutable Date accessed;
  };
  // ...
};
```

C++17:
http://coliru.stacked-crooked.com/a/e8eddd01f177a572

Essence or Accident? C++20

```
template<class K, class V, size_t SIZE, class FetchFunc>
class Cache {
  struct Holder {
    V val;
    mutable ~~typename~~ list<K>::iterator posInList;
    Holder(V v, Date exp, ~~typename~~ list<K>::iterator pos)
    : val(v), posInList(pos), expiry(exp), accessed(Date()) {}
    // ...
  private:
    Date expiry;
    mutable Date accessed;
  };
  // ...
};
```

C++20:
http://coliru.stacked-crooked.com/a/47ce82fa46ffe3ba

See: Why don't I need to specify "typename" before a dependent type in C++20?
And: Why is `typename` prefix still required in such a case in C++20?

Essence or Accident?

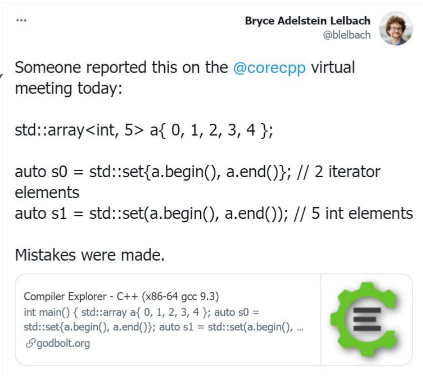
Essence or Accident?

```cpp
int main() {
  int arr[] = {1, 2, 3, 3, 2, 1};
  std::set unique_values{std::begin(arr), std::end(arr)};
  for(auto val : unique_values) {
    std::cout << val << ' ';
  }
}
```

Essence or Accident?

```cpp
int main() {
  int arr[] = {1, 2, 3, 3, 2, 1};
  std::set unique_values{std::begin(arr), std::end(arr)};
  for(auto val : unique_values) {
    std::cout << val << ' ';
  }
}
```

Someone = Andrei Zissu



Bryce Adelstein Lelbach
@blelbach

Someone reported this on the @corecpp virtual meeting today:

std::array<int, 5> a{ 0, 1, 2, 3, 4 };

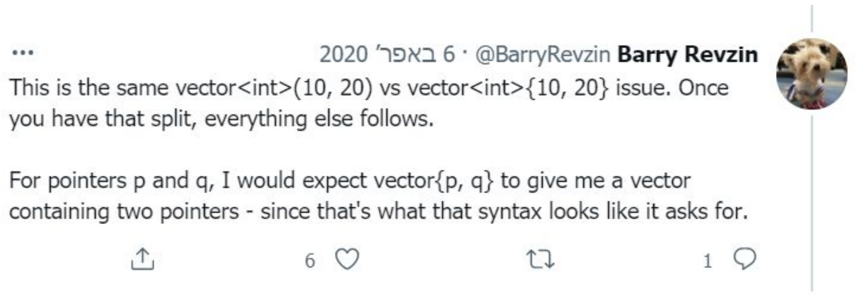
auto s0 = std::set{a.begin(), a.end()}; // 2 iterator elements
auto s1 = std::set(a.begin(), a.end()); // 5 int elements

Mistakes were made.

Compiler Explorer - C++ (x86-64 gcc 9.3)
int main() { std::array a{ 0, 1, 2, 3, 4 }; auto s0 = std::set{a.begin(), a.end()}; auto s1 = std::set(a.begin(), ...
godbolt.org

Essence or Accident?

```
int main() {
  int arr[] = {1, 2, 3, 3, 2, 1};
  std::set unique_values{std::begin(arr), std::end(arr)};
  for(auto val : unique_values) {
    std::cout << val << ' ';
  }
}
```



2020 באפר׳ 6 · @BarryRevzin **Barry Revzin**

This is the same vector<int>(10, 20) vs vector<int>{10, 20} issue. Once you have that split, everything else follows.

For pointers p and q, I would expect vector{p, q} to give me a vector containing two pointers - since that's what that syntax looks like it asks for.

6 1

Essence or Accident?

How to store a value obtained from a vector `pop_back()`?

Essence or Accident?

How to store a value obtained from a vector `pop_back()`?

```
auto val = vec.back();
vec.pop_back();
```

^ Maybe?

Essence or Accident?

How to store a value obtained from a vector `pop_back()`?

```
auto val = vec.back();
vec.pop_back();
```

^ Maybe? Not really...

Essence or Accident?

How to store a value obtained from a vector `pop_back()`?

```
auto val = std::move(vec.back());
vec.pop_back();
```

Essence or Accident?

Essence or Accident?

```
std::vector<bool> flags;
// ...

// need to toggle all flags
for(auto&  flag: flags) {
  flag = !flag;
}
```

Essence or Accident?

```
std::vector<bool> flags;
// ...

// need to toggle all flags
for(auto&& flag: flags) {
  flag = !flag;
}
```

Essence or Accident?

Essence or Accident?

```
struct A {
  int foo(int) { return 7; }
};

struct B: A {
  int foo(float) { return 8; }
};

int main() {
  return B().foo(0); // 8 or 7 ?
}
```

Essence or Accident?

Essence or Accident?

```
template<typename T>
std::enable_if_t<std::is_integral_v<T>> f(T t) {
   // integral version
}

template<typename T>
std::enable_if_t<std::is_floating_point_v<T>> f(T t) {
   // floating point version
}
```

Essence or Accident?

```
template<typename T>
std::enable_if_t<std::is_integral_v<T>> f(T t) {
   // integral version
}

template<typename T>
std::enable_if_t<std::is_floating_point_v<T>> f(T t) {
   // floating point version
}
```

⇩ C++20

```
void f(std::integral auto t) {          void f(std::floating_point auto t) {
   //integral version                      //floating point version
}                                       }
```

Essence or Accident?

Essence or Accident?

```
std::string s = "but I have heard it works even if you don’t believe in it";
s.replace(0, 4, "").replace(s.find("even"), 4, "only").replace(s.find(" don’t"), 6, "");
assert(s == "I have heard it works only if you believe in it");
```

Essence or Accident?

```
std::string s = "but I have heard it works even if you don’t believe in it";
s.replace(0, 4, "").replace(s.find("even"), 4, "only").replace(s.find(" don’t"), 6, "");
assert(s == "I have heard it works only if you believe in it");
```

Chaining is fixed, but only since C++17:
http://open-std.org/JTC1/SC22/WG21/docs/papers/2016/p0145r3.pdf

Very relevant to the pipe | syntax used by ranges

C++ Principles (Stroustrup, C++ Design and Evolution)

```
Static type system
- equal support for builtins and user
  defined types
- value and reference semantics

Resource and Memory management
- RAII - scoped based
- No garbage collector

Efficient Object Oriented Programming

Flexible and efficient generic
programming

Pay only for what you need

Direct access to OS and HW

Leave no room for a lower-level
language below C++
* except assembler

See:
```

The Design of C++, by Bjarne Stroustrup, 1994

The Acronyms

The Acronyms - partial list

ODR IFNDR IILE CTAD RAII ADL RVO UB EBO CWG LTO CRTP RTTI TU SFINAE UDL

The Acronyms - partial list

ODR IFNDR IILE CTAD

It's not complex… just go to the C++ acronym glossary by Arthur O'Dwyer

CWG LTO CRTP RTTI TU SFINAE UDL

The Acronyms - partial list

ODR IFNDR

IILE CTAD

Or join Bob Steagall’s talk here at CppCon 2021 on Friday afternoon

CWG LTO CRTP RTTI

TU

SFINAE UDL

The Acronyms - partial list

ODR IFNDR

IILE CTAD

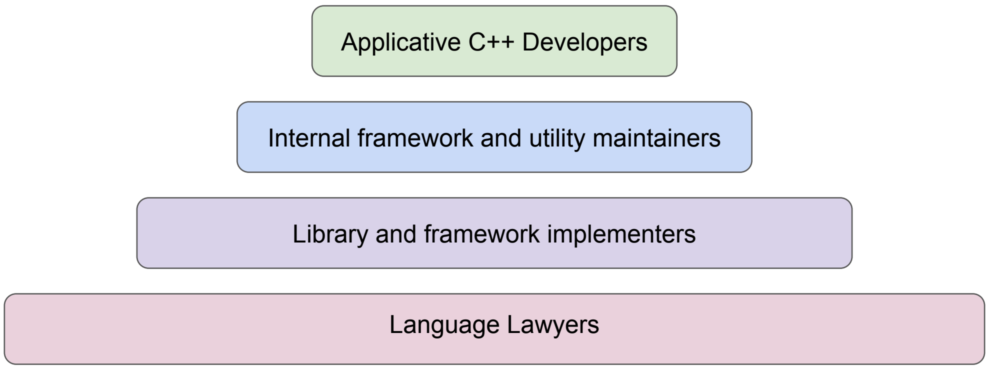
Watch also Kate Gregory’s great talk “It’s Complicated” from Meeting C++17

CWG LTO CRTP RTTI

TU

SFINAE UDL

The Pyramid of C++ Knowledge


Applicative C++ Developers
Internal framework and utility maintainers
Library and framework implementers
Language Lawyers

The bare minimum to be a C++ programmer

The bare minimum to be a C++ programmer

the basic syntax, implicit casting rules, const correctness, constexpr,
RAII, Rule of Zero, Rule of Three, operators overloading,
static variables and static members,
RValue and move semantics, Rule of Five,
inheritance, polymorphism, multiple inheritance, virtual inheritance,
exceptions, basic templates, variadic templates,
forwarding reference and perfect forwarding,
std containers, std algorithms, function objects, lambda,
use of smart pointers

The bare minimum to be a C++ programmer

the basic syntax, implicit casting rules, const correctness, constexpr,
RAII, Rule of Zero, Rule of Three, operators overloading,
static variables and static members,
RValue and move semantics, Rule of Five,
inheritance, polymorphism, ~~multiple inheritance~~, ~~virtual inheritance~~,
exceptions, basic templates, ~~variadic templates~~,
~~forwarding reference and perfect forwarding~~,
std containers, std algorithms, function objects, lambda,
use of smart pointers

The bare minimum to be a C++ programmer

the basic syntax, implicit casting rules, const correctness, constexpr,
RAII, Rule of Zero, Rule of Three, operators overloading,
static variables and static members,
RValue and move semantics, Rule of Five,
inheritance, polymorphism, ~~multiple inheritance~~, ~~virtual inheritance~~,
exceptions, basic templates, ~~variadic templates~~,
~~forwarding reference and perfect forwarding~~,
std containers, std algorithms, function objects, lambda,
use of smart pointers

reading code

browsing cppreference and stackoverflow

Being able to read is important, even crucial

Being able to read C++ code is even more important than writing

- know what you know
- know what you don’t know
- learn

Interviewing for a C++ junior position

Interviewing for a C++ junior position

```
Knows the bare minimum, or we are ready to train.
```

Interviewing for a C++ junior position

Knows the bare minimum, or we are ready to train.

Loves programming. Really, loves programming!

Interviewing for a C++ junior position

Knows the bare minimum, or we are ready to train.

Loves programming. Really, loves programming!

Smart and gets things done.

Interviewing for a C++ junior position

Knows the bare minimum, or we are ready to train.

Loves programming. Really, loves programming!

Smart and gets things done.

SMART & GETS THINGS DONE
JOEL SPOLSKY's CONCISE GUIDE TO FINDING THE BEST TECHNICAL TALENT

Implications of innocent ignorance

Well, I didn’t know that…

Implications of innocent ignorance

`Well, I didn’t know that…`

- `Less elegant code (harder to maintain, harder to read)`
- `Less Generic code (could be written in a more generic way)`
- `Not being able to implement things`
- `Inefficient code`
- `Bug prone`
- `Actual bug!`

Implications of innocent ignorance

`Well, I didn’t know that…`

- `Less elegant code (harder to maintain, harder to read)`
- `Less Generic code (could be written in a more generic way)`
- `Not being able to implement things`
- `Inefficient code`
- `Bug prone`
- `Actual bug!`

⬆ still positive value

⬇ negative value

Improving *your* C++ level

Improving *your* C++ level

```
Curiosity - in C++ everything has a reason, try to figure it out
```

Improving *your* C++ level

Curiosity - in C++ everything has a reason, try to figure it out

Read, listen, watch

Improving *your* C++ level

Curiosity - in C++ everything has a reason, try to figure it out

Read, listen, watch

^ Read Q&A in Stackoverflow

Improving *your* C++ level

Curiosity - in C++ everything has a reason, try to figure it out

Read, listen, watch

^ Read Q&A in Stackoverflow

^ Ask in Stackoverflow

Improving *your* C++ level

Curiosity - in C++ everything has a reason, try to figure it out

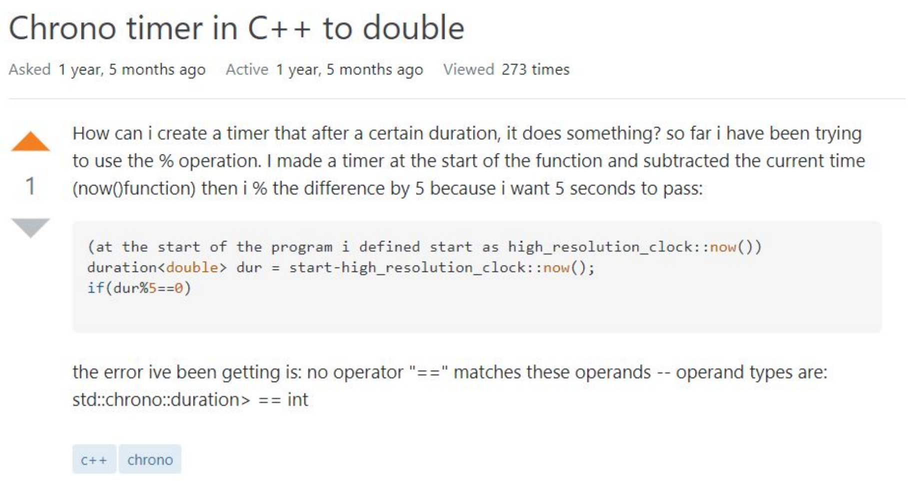
Read, listen, watch

^ Read Q&A in Stackoverflow

^ Ask in Stackoverflow

^ Answer in Stackoverflow

Improving while answering in SO



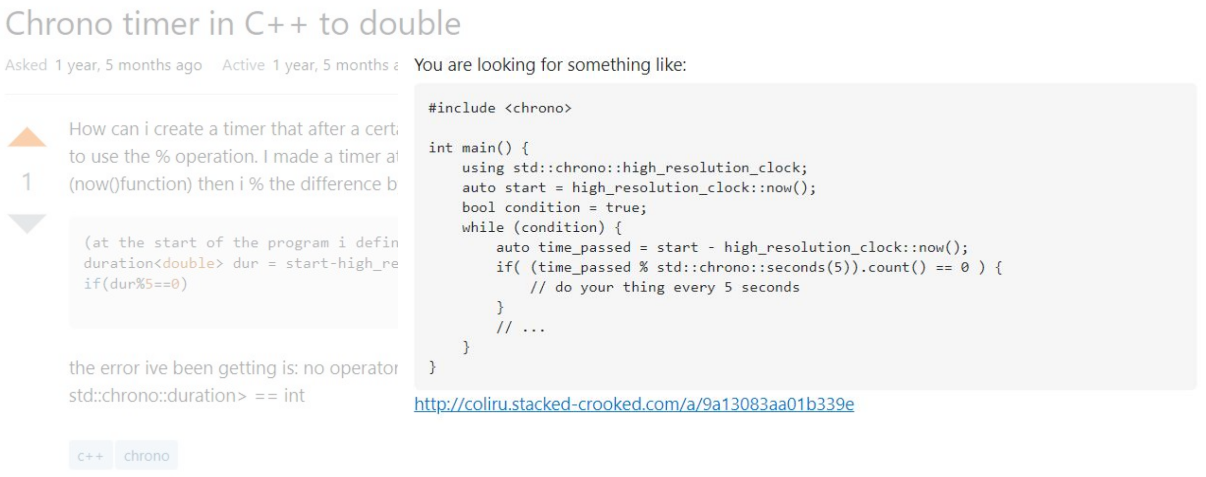
Chrono timer in C++ to double

Asked 1 year, 5 months ago Active 1 year, 5 months ago Viewed 273 times

How can i create a timer that after a certain duration, it does something? so far i have been trying to use the % operation. I made a timer at the start of the function and subtracted the current time (now()function) then i % the difference by 5 because i want 5 seconds to pass:

```
(at the start of the program i defined start as high_resolution_clock::now())
duration<double> dur = start-high_resolution_clock::now();
if(dur%5==0)
```

the error ive been getting is: no operator "==" matches these operands -- operand types are: std::chrono::duration> == int



Chrono timer in C++ to double

Asked 1 year, 5 months ago Active 1 year, 5 months a

1

How can i create a timer that after a cert
to use the % operation. I made a timer at
(now()function) then i % the difference b

```
(at the start of the program i defin
duration<double> dur = start-high_re
if(dur%5==0)
```

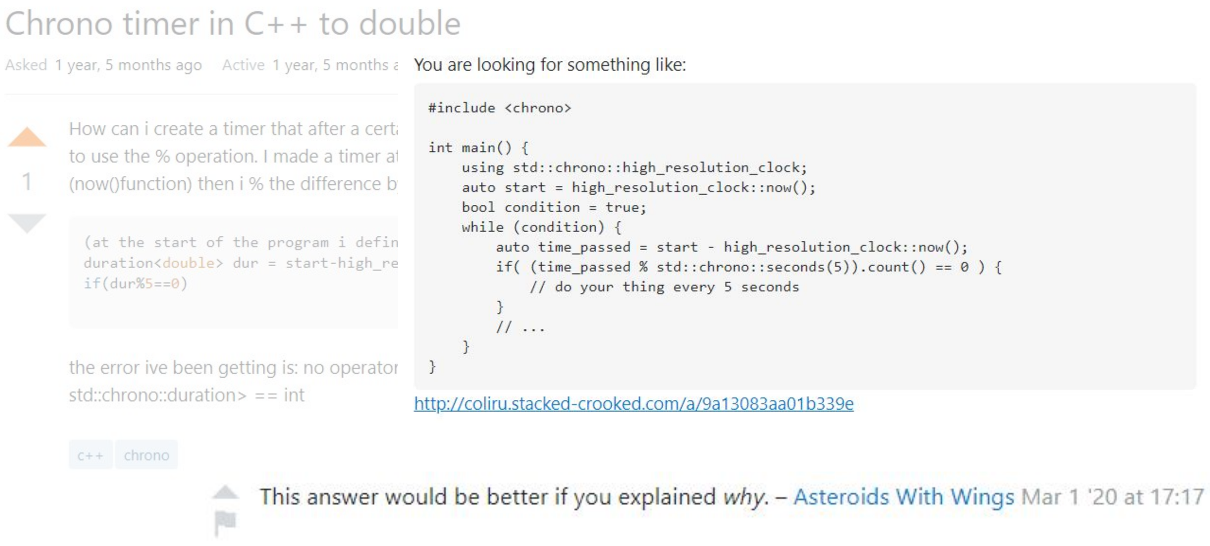
the error ive been getting is: no operator
std::chrono::duration> == int

c++ chrono

You are looking for something like:

```
#include <chrono>

int main() {
    using std::chrono::high_resolution_clock;
    auto start = high_resolution_clock::now();
    bool condition = true;
    while (condition) {
        auto time_passed = start - high_resolution_clock::now();
        if( (time_passed % std::chrono::seconds(5)).count() == 0 ) {
            // do your thing every 5 seconds
        }
        // ...
    }
}
```

http://coliru.stacked-crooked.com/a/9a13083aa01b339e



Chrono timer in C++ to double

Asked 1 year, 5 months ago Active 1 year, 5 months a

1

How can i create a timer that after a cert
to use the % operation. I made a timer at
(now()function) then i % the difference b

```
(at the start of the program i defin
duration<double> dur = start-high_re
if(dur%5==0)
```

the error ive been getting is: no operator
std::chrono::duration> == int

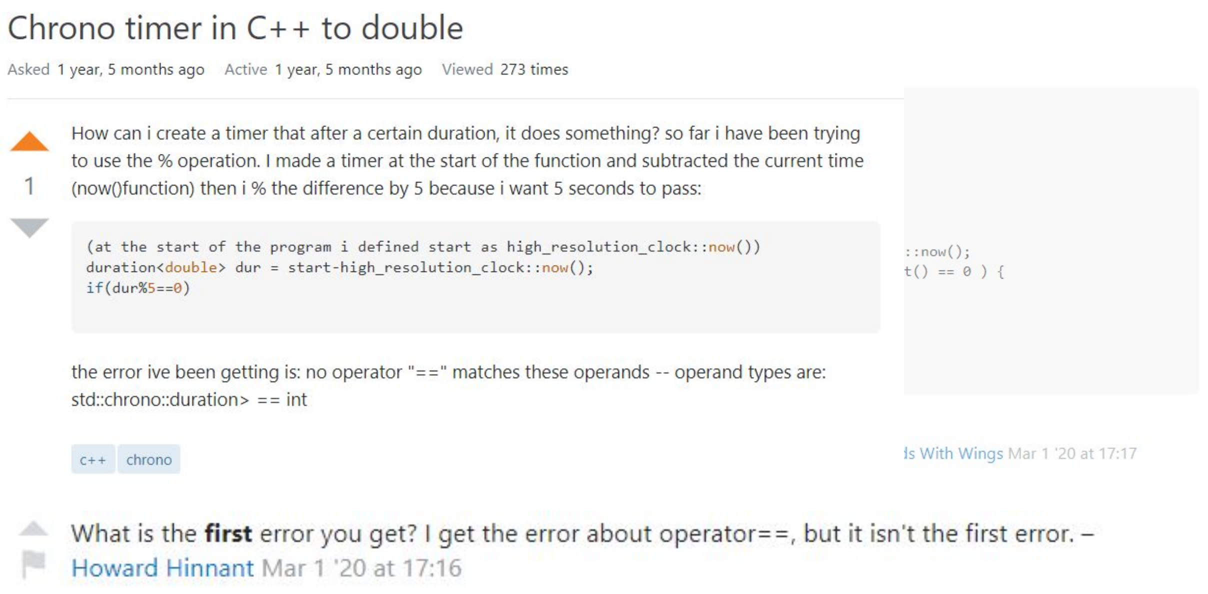
c++ chrono

You are looking for something like:

```
#include <chrono>

int main() {
    using std::chrono::high_resolution_clock;
    auto start = high_resolution_clock::now();
    bool condition = true;
    while (condition) {
        auto time_passed = start - high_resolution_clock::now();
        if( (time_passed % std::chrono::seconds(5)).count() == 0 ) {
            // do your thing every 5 seconds
        }
        // ...
    }
}
```

http://coliru.stacked-crooked.com/a/9a13083aa01b339e

This answer would be better if you explained *why*. – Asteroids With Wings Mar 1 '20 at 17:17



Chrono timer in C++ to double

Asked 1 year, 5 months ago Active 1 year, 5 months ago Viewed 273 times

1

How can i create a timer that after a certain duration, it does something? so far i have been trying to use the % operation. I made a timer at the start of the function and subtracted the current time (now()function) then i % the difference by 5 because i want 5 seconds to pass:

```
(at the start of the program i defined start as high_resolution_clock::now())
duration<double> dur = start-high_resolution_clock::now();
if(dur%5==0)
```

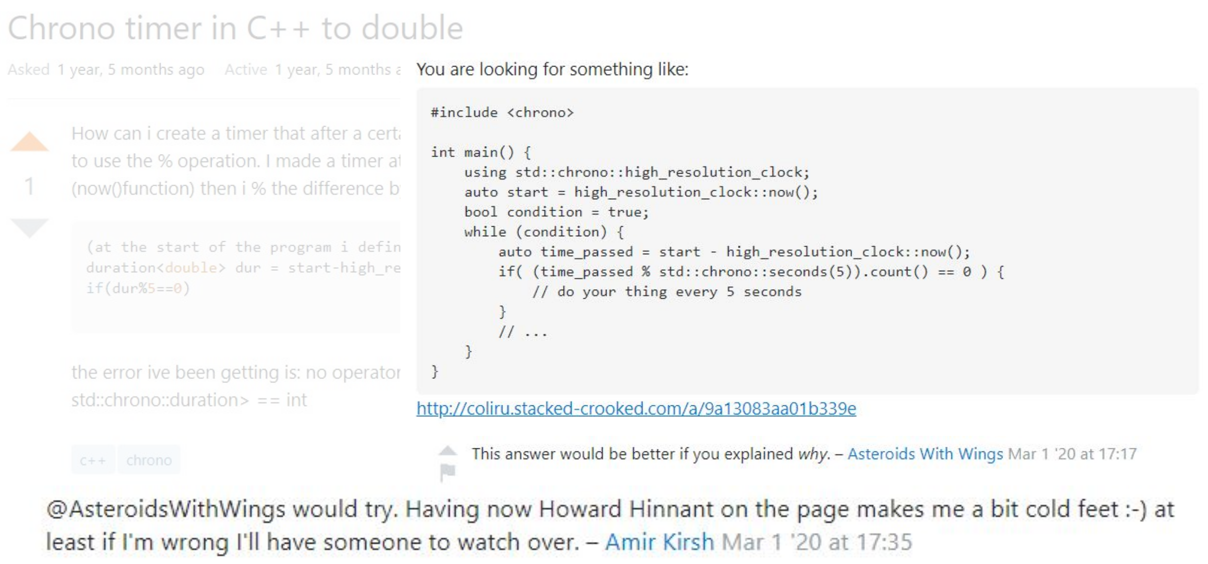
the error ive been getting is: no operator "==" matches these operands -- operand types are: std::chrono::duration> == int

c++ chrono

```
::now();
t() == 0 ) {
```

ds With Wings Mar 1 '20 at 17:17

What is the **first** error you get? I get the error about operator==, but it isn't the first error. – Howard Hinnant Mar 1 '20 at 17:16



Chrono timer in C++ to double

Asked 1 year, 5 months ago Active 1 year, 5 months a

1

How can i create a timer that after a cert
to use the % operation. I made a timer a
(now()function) then i % the difference b

```
(at the start of the program i defin
duration<double> dur = start-high_re
if(dur%5==0)
```

the error ive been getting is: no operator
std::chrono::duration> == int

c++ chrono

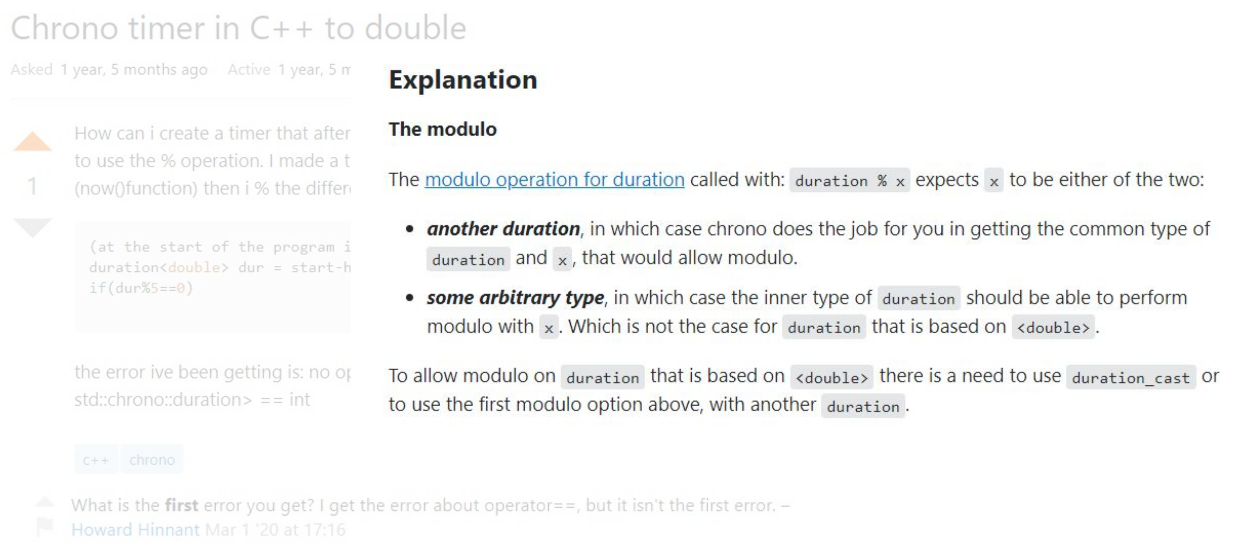
You are looking for something like:

```
#include <chrono>

int main() {
    using std::chrono::high_resolution_clock;
    auto start = high_resolution_clock::now();
    bool condition = true;
    while (condition) {
        auto time_passed = start - high_resolution_clock::now();
        if( (time_passed % std::chrono::seconds(5)).count() == 0 ) {
            // do your thing every 5 seconds
        }
        // ...
    }
}
```

http://coliru.stacked-crooked.com/a/9a13083aa01b339e

This answer would be better if you explained *why*. – Asteroids With Wings Mar 1 '20 at 17:17

@AsteroidsWithWings would try. Having now Howard Hinnant on the page makes me a bit cold feet :-) at least if I'm wrong I'll have someone to watch over. – Amir Kirsh Mar 1 '20 at 17:35



Chrono timer in C++ to double

Asked 1 year, 5 months ago Active 1 year, 5 m

1

How can i create a timer that after
to use the % operation. I made a t
(now()function) then i % the differ

```
(at the start of the program i
duration<double> dur = start-h
if(dur%5==0)
```

the error ive been getting is: no op
std::chrono::duration> == int

c++ chrono

What is the **first** error you get? I get the error about operator==, but it isn't the first error. – Howard Hinnant Mar 1 '20 at 17:16

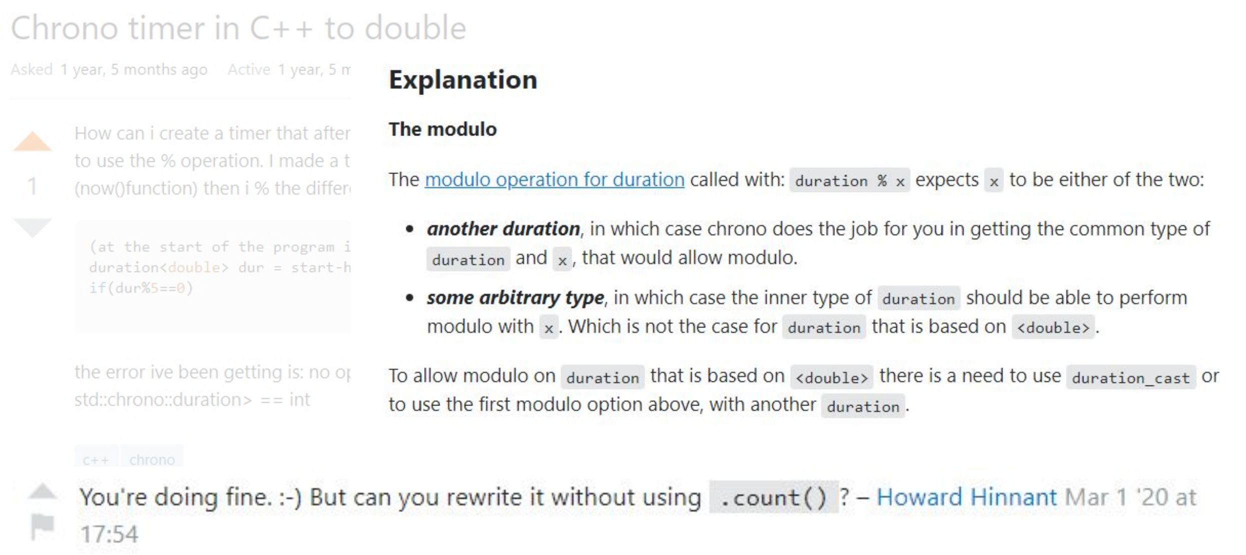
Explanation

The modulo

The modulo operation for duration called with: `duration % x` expects `x` to be either of the two:

- ***another duration***, in which case chrono does the job for you in getting the common type of `duration` and `x`, that would allow modulo.
- ***some arbitrary type***, in which case the inner type of `duration` should be able to perform modulo with `x`. Which is not the case for `duration` that is based on `<double>`.

To allow modulo on `duration` that is based on `<double>` there is a need to use `duration_cast` or to use the first modulo option above, with another `duration`.



Chrono timer in C++ to double

Asked 1 year, 5 months ago Active 1 year, 5 m

1

How can i create a timer that after
to use the % operation. I made a t
(now()function) then i % the differ

```
(at the start of the program i
duration<double> dur = start-h
if(dur%5==0)
```

the error ive been getting is: no op
std::chrono::duration> == int

c++ chrono

Explanation

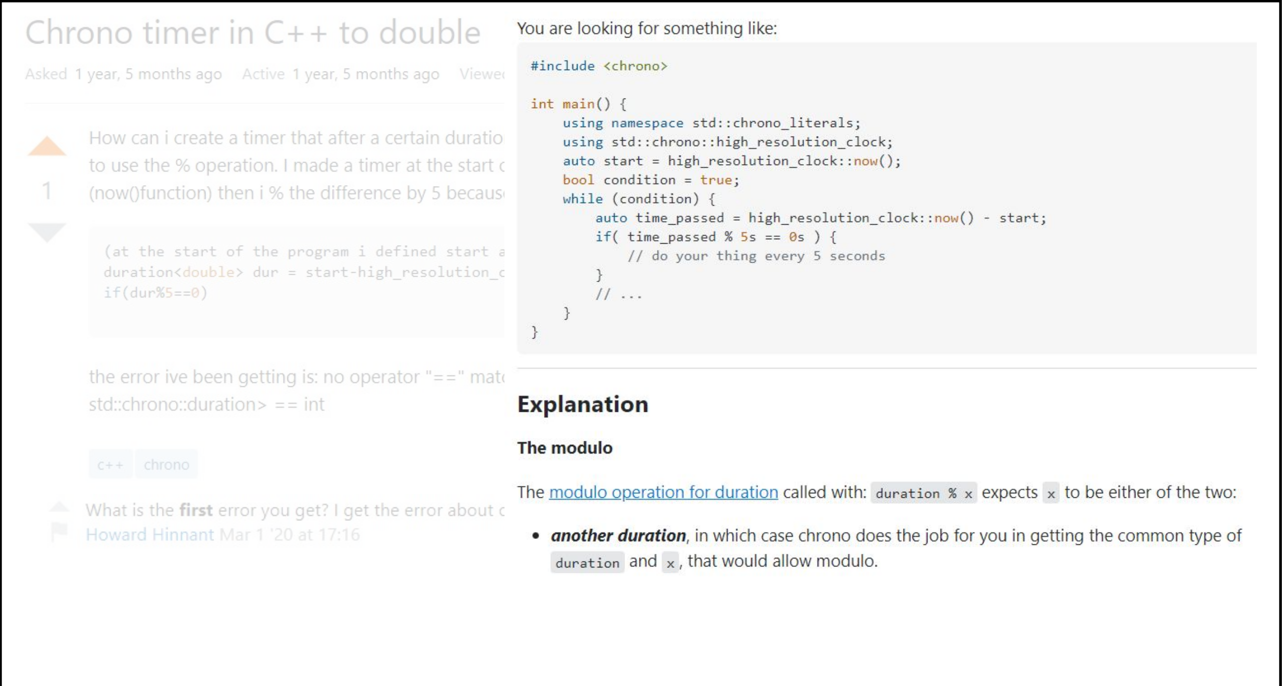
The modulo

The modulo operation for duration called with: `duration % x` expects `x` to be either of the two:

- ***another duration***, in which case chrono does the job for you in getting the common type of `duration` and `x`, that would allow modulo.
- ***some arbitrary type***, in which case the inner type of `duration` should be able to perform modulo with `x`. Which is not the case for `duration` that is based on `<double>`.

To allow modulo on `duration` that is based on `<double>` there is a need to use `duration_cast` or to use the first modulo option above, with another `duration`.

You're doing fine. :-) But can you rewrite it without using `.count()`? – Howard Hinnant Mar 1 '20 at 17:54



Chrono timer in C++ to double

Asked 1 year, 5 months ago Active 1 year, 5 months ago Viewe

1

How can i create a timer that after a certain duratio
to use the % operation. I made a timer at the start c
(now()function) then i % the difference by 5 becaus

```
(at the start of the program i defined start a
duration<double> dur = start-high_resolution_c
if(dur%5==0)
```

the error ive been getting is: no operator "==" mat
std::chrono::duration> == int

c++ chrono

What is the **first** error you get? I get the error about c
Howard Hinnant Mar 1 '20 at 17:16

You are looking for something like:

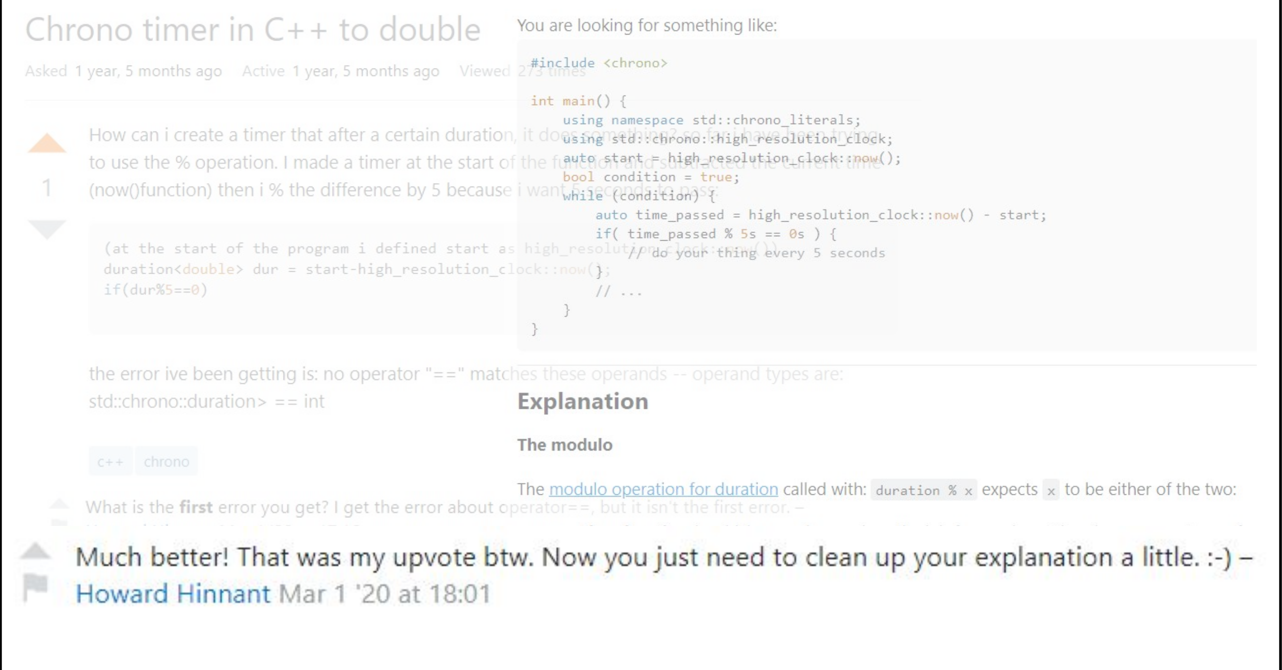
```
#include <chrono>

int main() {
    using namespace std::chrono_literals;
    using std::chrono::high_resolution_clock;
    auto start = high_resolution_clock::now();
    bool condition = true;
    while (condition) {
        auto time_passed = high_resolution_clock::now() - start;
        if( time_passed % 5s == 0s ) {
            // do your thing every 5 seconds
        }
        // ...
    }
}
```

Explanation

The modulo

The modulo operation for duration called with: `duration % x` expects `x` to be either of the two:

- ***another duration***, in which case chrono does the job for you in getting the common type of `duration` and `x`, that would allow modulo.



Chrono timer in C++ to double

Asked 1 year, 5 months ago Active 1 year, 5 months ago Viewed 273 times

1

How can i create a timer that after a certain duration, it does something? so far i have been trying to use the % operation. I made a timer at the start of the function and subtracted the current time (now()function) then i % the difference by 5 because i want 5 seconds to pass:

```
(at the start of the program i defined start as high_resolution_clock::now())
duration<double> dur = start-high_resolution_clock::now();
if(dur%5==0)
```

the error ive been getting is: no operator "==" matches these operands -- operand types are: std::chrono::duration> == int

c++ chrono

What is the **first** error you get? I get the error about operator==, but it isn't the first error. –

You are looking for something like:

```
#include <chrono>

int main() {
    using namespace std::chrono_literals;
    using std::chrono::high_resolution_clock;
    auto start = high_resolution_clock::now();
    bool condition = true;
    while (condition) {
        auto time_passed = high_resolution_clock::now() - start;
        if( time_passed % 5s == 0s ) {
            // do your thing every 5 seconds
        }
        // ...
    }
}
```

Explanation

The modulo

The modulo operation for duration called with: `duration % x` expects `x` to be either of the two:

Much better! That was my upvote btw. Now you just need to clean up your explanation a little. :-) – Howard Hinnant Mar 1 '20 at 18:01

Improving *your* C++ level

Remember, it’s a never ending mission

Thank you!

```
void conclude(auto greetings) {
    while(still_time() && have_questions()) {
        ask();
    }
    greetings();
}

conclude([]{ std::cout << "Thank you!"; });
```

Other Essence or Accident

out due to lack of time

Essence or Accident?

```
template<class T, long Numerator, long Denominator, long MultNum, long MultDenom>
auto constexpr operator*(Aggregator<T, Numerator, Denominator> a,
                         std::ratio<MultNum, MultDenom> n) {
  if constexpr(Numerator*MultNum != Denominator*MultDenom) {
    return Aggregator<T, Numerator * MultNum, Denominator * MultDenom> { a };
  } else {
    return a.unsafe_multiply(n);
  }
}
```

Source: The Point Challenge https://www.youtube.com/watch?v=wNGEtlBSCLY

Essence or Accident?

Implement methods for rotating the x,y,z fields in Pixel struct below:

```cpp
struct Pixel {
   int x;
   int y;
   int z;
};
```

Essence or Accident?

```cpp
class Permutation {
   std::array<int Pixel::*, 3> permutation;
   constexpr Permutation(int Pixel::* a, int Pixel::* b, int Pixel::* c)
       : permutation{a, b, c} {}
public:
   static constexpr Permutation xzy() { return {&Pixel::x, &Pixel::z, &Pixel::y}; }
   constexpr Pixel permute(Pixel p) const {
       Pixel permutated;
       permutated.x = p.*permutation[0];
       permutated.y = p.*permutation[1];
       permutated.z = p.*permutation[2];
       return permutated;
   }
};
```

http://coliru.stacked-crooked.com/a/298a6e5a89e10a28