

Department I - C Plus Plus

Modern and Lucid C++ Advanced for Professional Programmers

Week 0 – Introduction

Thomas Corbat / Felix Morgner
Rapperswil, 17.02.2020
FS2020



- **4 Credits (2h Vorlesung / 2h Übungen pro Woche)**

- **Testate**

- Gruppenarbeit 2-3 Studierende
- 2 von 3 Abgaben müssen bestanden sein (Zeitpunkt wird bei Bedarf noch angepasst)
- Keine Note
- Zulassungsbedingung für die Prüfung

- **Prüfung**

- 120 Minuten
- Open Book (Ausnahme alte Prüfungen oder Abschriften davon)

- **Neu: Alle Unterlagen auf Gitlab**

- <https://gitlab.dev.ifs.hsr.ch/tcorbat/cppadvanced>

- Account kann mit HSR-Mailadresse erstellt werden

- **Austausch von nicht öffentlichen Inhalten über Skripte-Server:**

- https://skripte.hsr.ch/Informatik/Fachbereich/C++_Advanced/CplA/

- \\hsr.ch\root\alg\skripte\Informatik\Fachbereich\C++_Advanced\CplA

- **Arbeitsumgebung (Vorläufig)**

- Aktueller Compiler (GCC 9.2.0+): Windows MinGW von <https://nuwen.net/mingw.html>

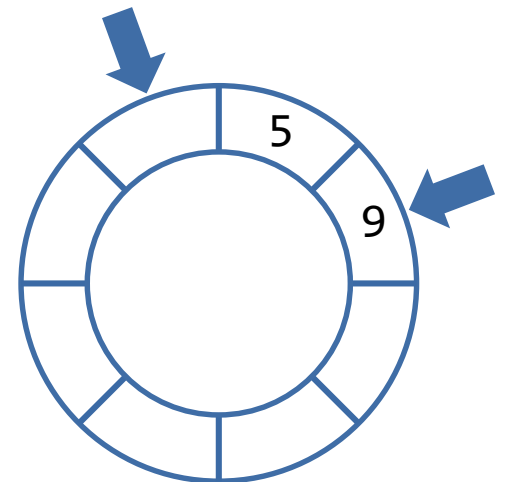
- Cevalop Version 1.14.0 (www.cevelop.com)

- **An Ostermontag kein Unterricht: Montag 13.04.2020**

- **What is your motivation?**
- **What are your expectations?**

- **Short refresh of C++ topics (Today)**
- **Move semantics and type deduction**
- **Low level memory management**
- **Compile-time C++**
- **Multi-Threading / Memory Model**
- **Networking**
- **Advanced Libraries/Development**
- **Whole Value Types**
- **Core Guidelines**
- **Build Systems**

- **Typical tasks known from C++**
- **One specific exercise that will be part of several exercises**
 - Bounded Buffer
 - Start with a simple `std::array`-based version that will be extended during the semester
 - Parts of it will be Testat



- **Get Cevelop**

- at <https://www.cevelop.com/download>

- **With plug-ins Constificator and Stylechecker preinstalled**

- For additional plug-ins (already configured in Cevelop): <https://www.cevelop.com/update/latest>

Cevelop

Home

News

Download

Features

FAQ

Cevelop

Download – the latest release of Cevelop

Download Cevelop for your preferred operating system

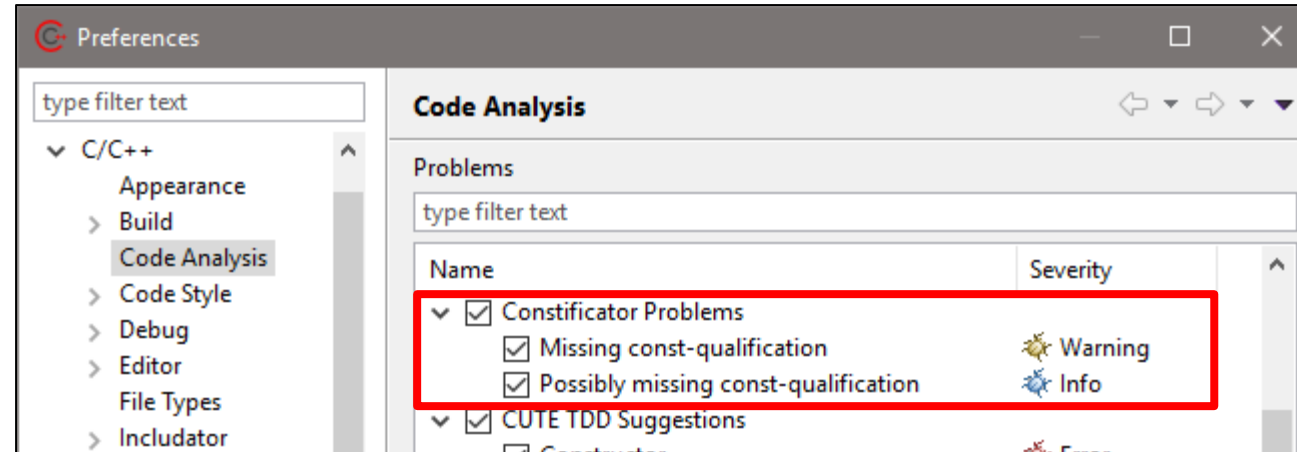
 Windows

 OS X

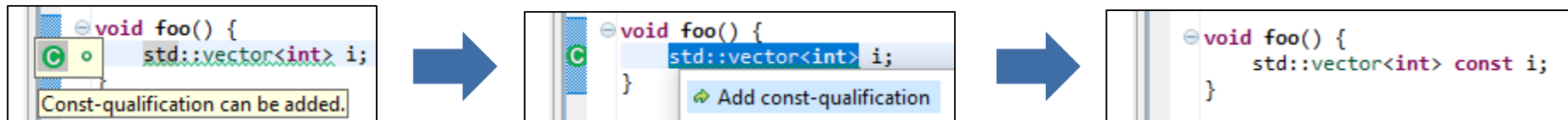
 Linux

● Constificator

- If activated



- Makes suggestions about missing const qualifiers



- **It annoys you in the first place!**

- When you have multiple ASSERTs in your test cases

```
void thisIsATest() {  
    ASSERTM("start writing tests", false);  
    ASSERTM("Only one ASSERT per test", true);  
}
```

A test should only contain one assert.

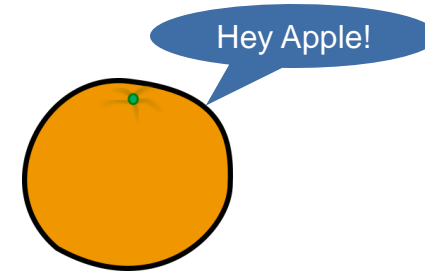
- When you use std::cin/std::cout outside of the main function

```
void foo() {  
    std::cout << "output";  
}
```

Cin/Cout should only be used inside the main function

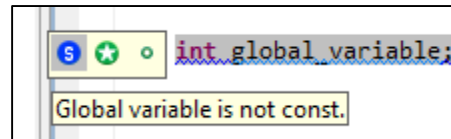
- When there are issues with your include guard

```
#ifndef SRC_HEADER_H  
#define SRC_HEADER_H  
Header files should have an include guard for symbol HEADER_H_  
#endif /* SRC_HEADER_H */
```

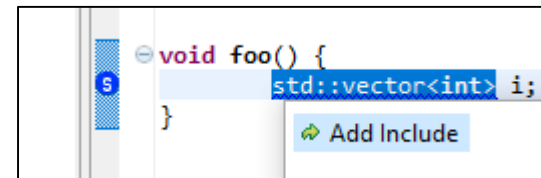
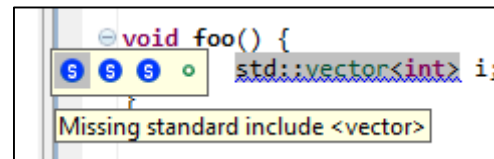


- and reports...

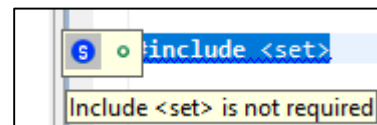
- ... global non-const variables



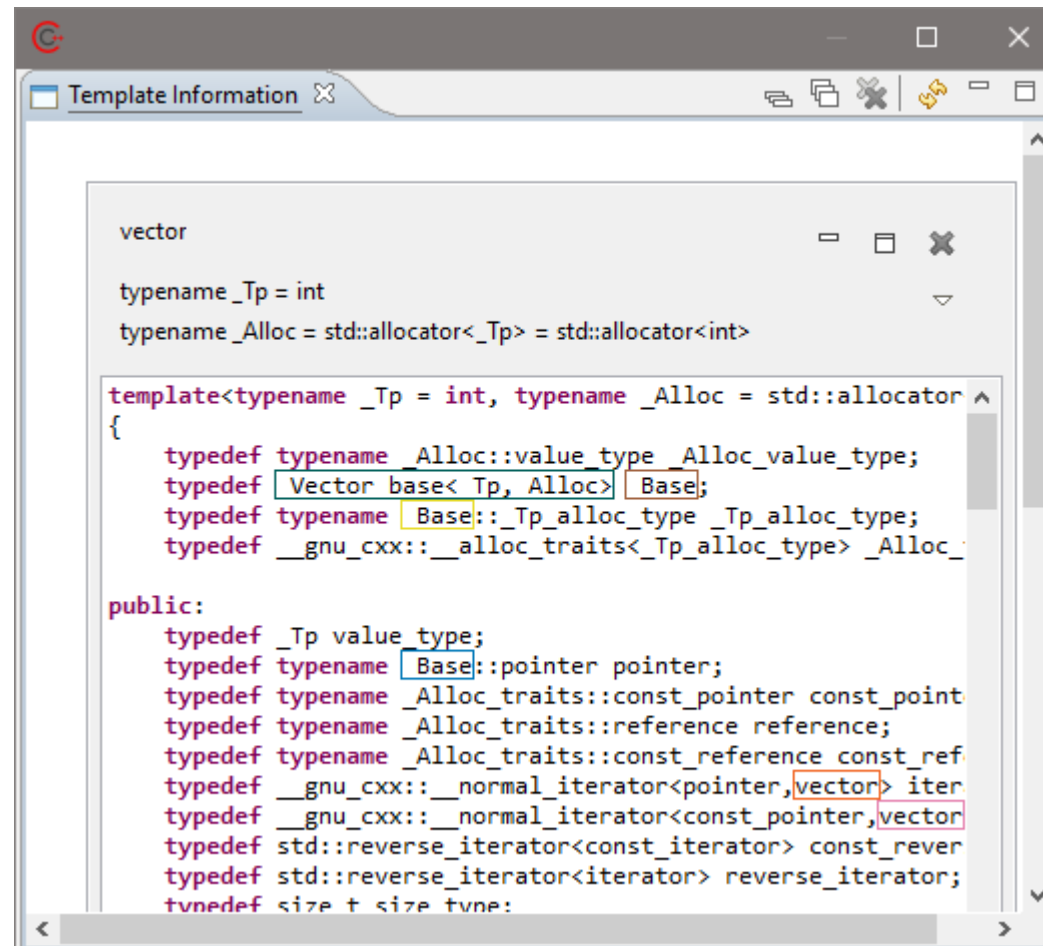
- ... missing includes of Standard Library headers and provides a resolution!



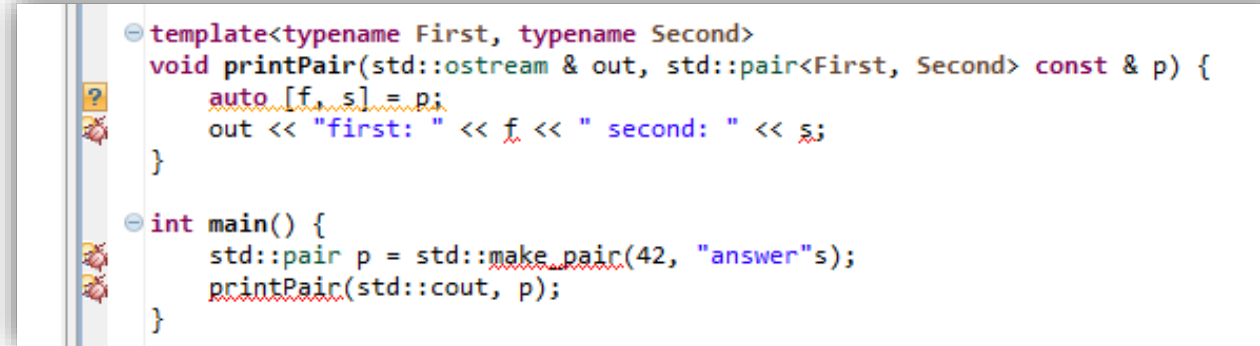
- ... unused includes of Standard Library headers



- Template Instance Visualization



- You might encounter several bugs in Cevalop and its plug-ins (more than in CPI)
 - We are using modern and sophisticated features of C++17, which are not completely supported yet



```
template<typename First, typename Second>
void printPair(std::ostream & out, std::pair<First, Second> const & p) {
    auto [f, s] = p;
    out << "first: " << f << " second: " << s;
}

int main() {
    std::pair p = std::make_pair(42, "answer"s);
    printPair(std::cout, p);
}
```

- When in doubt ask your compiler
- When in more doubt ask several compilers
 - <http://melpon.org/wandbox>
 - <https://gcc.godbolt.org/>