



STATIC ASSERTION



MOTIVATION

- Keine Laufzeit-Performance-Einbußen
- Kein Speicherplatz benötigt
- Aussagekräftige Fehlermeldungen

VORHERIGE LÖSUNGSANSÄTZE

- `assert(index >= 0 && index <= 9);`
- Assertion failed: index >= 0 && index <=9
- `assert(found && "Car could not be found in database");`
- Assertion failed: found && "Car could not be found in database"

SYNTAX

Ab C++ 11:

- `static_assert (bool_constexpr , message)`

Ab C++ 17:

- `static_assert (bool_constexpr)`
-

CODE BEISPIELE

- Block Scope

```
int main() {  
    static_assert(sizeof(int) == sizeof(long int), "int and long int must be of the same length.");  
}
```

WEITERE CODEBEISPIELE

- Class Scope

```
template <class T, int Size>
class Vector { static_assert(Size < 3, "Vector size is too small!"); T m_values[Size]; };

int main() {
    Vector<int, 4> four; // This will work
    Vector<short, 2> two; // This will fail
    return 0; }
```

- Ausgabe: *error: static assertion failed:Vector size is too small!*

- Namespace Scope (global):

```
static_assert(sizeof(void *) == 4, "64-bit code generation is not supported.");
```