101 brilliant things of C++

Andreas Fertig
https://AndreasFertig.Info
post@AndreasFertig.Info
@Andreas_Fertig

fertig

adjective /ˈfɛrtɪç/

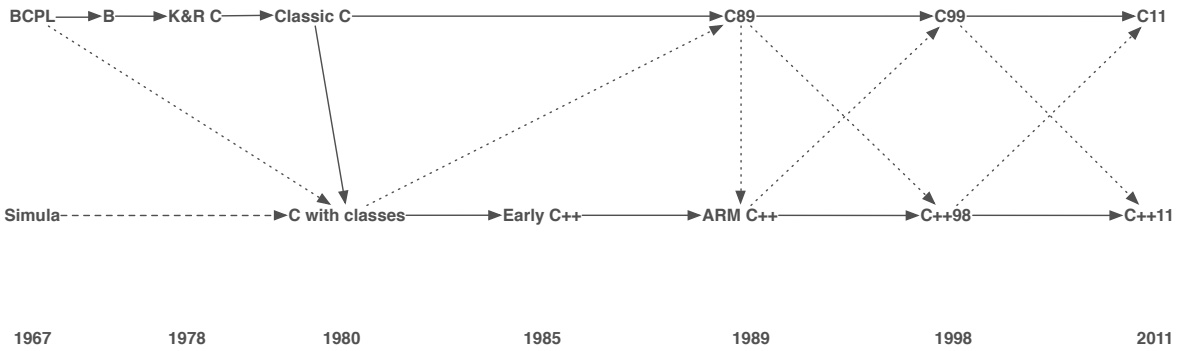
finished
ready
complete
completed

101 $\Rightarrow$ 5

The History of C++

- Developed 1979 by Bjarne Stroustrup at Bell Labs.
- His intention was to create a flexible language like C but with high-level features.
- The initial language was called *C with Classes*.
- C++ is a mix of different languages at the time Simula, BCPL, ALGOL 68, and ADA.
- C++ itself started around 1982 as a successor of C with Classes.
- First standardized in 1998, next time 2011, every three years since.
- Bjarne received the Charles Stark Draper Prize in 2018 for *conceptualizing and developing the C++ programming language*.

The History of C++



The History of C++

```
/* C-style comment */
// C++-style comment
```

Availability

Since 30+ years.
For probably every device out there you can find a C++ compiler.
ISO standard which you can help shaping.
Used in a wide variety of devices.
Large eco system.
Incredibly backward compatible.

Andreas Fertig
v1.0

101 brilliant things of C++ 7

Abstraction & Close to Hardware

$$\& \neq \&\&$$

$$| \neq ||$$

Binary Boolean

Andreas Fertig
v1.0

101 brilliant things of C++ 8

https://AndreasFertig.Info
post@AndreasFertig.Info

Abstraction & Close to Hardware

```
1 cout << ”Hello, C++!” << endl;
```


Abstraction & Close to Hardware

```
1 volatile uint8_t* usart0 =
2   reinterpret_cast<uint8_t*>(0x778AB);
3
4 *usart0 |= (1 << 4);
```

Abstraction & Close to Hardware

```
1 USART usart0{0x778AB};
2 usart0 |= UCSRnA::U2Xn;   A
3
4 usart0.Set(UCSRnA::U2Xn);   B
```


static type-safety

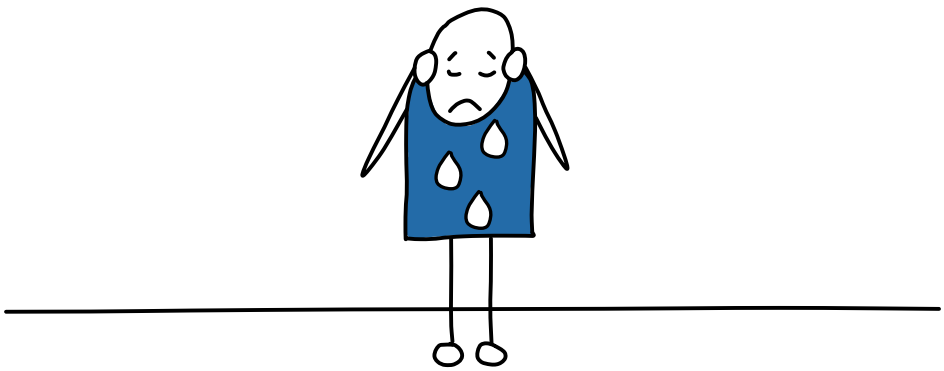
```
1  struct Apple {};
2
3  struct Orange {};
4
5  bool AreApplesOranges()
6  {
7    Apple  a{};
8    Orange o{};
9
10   return a == o;   A Does not compile
11 }
```

static type-safety

```
1  struct Apple {};
2
3  struct Orange {
4    bool operator==(const Apple&) const;
5  };
6
7  bool AreApplesOranges()
8  {
9    Apple  a{};
10   Orange o{};
11
12   return a == o;   A Compiles
13 }
```


static type-safety

```
1 double pi = 3.14;
2 int    a  = pi;
```



Generic Programming

"By generic programming we mean the definition of algorithms and data structures at an abstract or generic level, thereby accomplishing many related programming tasks simultaneously. The central notion is that of generic algorithms, which are parameterized procedural schemata that are completely independent of the underlying data representation and are derived from concrete, efficient algorithms."

— Musser and Stepanov [1]

Generic Programming

" Many programmers are unaware that C++ is more than an object-oriented language. C++ is also a language for generic programming, a methodology that can greatly enhance your ability to write efficient and reusable software components. "

— Austern [2]

Generic Programming

Standard Template Library (STL)

Generic Programming - Function Templates

```
1 template<typename T, int SIZE>
2 auto Fun(T param);
```

Example: `std::sort()`

https://AndreasFertig.Info
post@AndreasFertig.Info

Generic Programming - Class Templates

```
1 template<typename T, int SIZE>
2 class SomeClass;
```

Example: `std::vector<int>`

}

Destruktor

```
1  void ChangeScreen(Screen& newScreen)
2  {
3    lock();
4
5    if(screen == &newScreen) {
6      unlock();
7      return;
8    }
9
10   screen = &newScreen;
11   unlock();
12
13   SendUpdateNotificationEvent();
14 }
```

Andreas Fertig
v1.0

101 brilliant things of C++

21

Destruktor

```
1  struct Lock {
2    Lock() { lock(); }
3    ~Lock() { unlock(); };
4  };
5
6  void ChangeScreen(Screen& newScreen)
7  {
8    {
9      Lock lck{};
10
11     if(screen == &newScreen) { return; }
12
13     screen = &newScreen;
14   }
15
16   SendUpdateNotificationEvent();
17 }
```

Andreas Fertig
v1.0

101 brilliant things of C++

22

https://AndreasFertig.Info
post@AndreasFertig.Info

}

I am Fertig.

https://AndreasFertig.Info

Available online:

https://AndreasFertig.Info

Images by Franziska Panter:

https://panther-concepts.de

Used Compilers & Typography

Used Compilers

- Compilers used to compile (most of) the examples.
  - g++ 10.2.0
  - clang version 11.0.0 (https://github.com/llvm/llvm-project.git 176249bd6732a8044d457092ed932768724a6f06)

Typography

- Main font:
  - Camingo Dos Pro by Jan Fromm (https://janfromm.de/)
- Code font:
  - CamingoCode by Jan Fromm licensed under Creative Commons CC BY-ND, Version 3.0 http://creativecommons.org/licenses/by-nd/3.0/

References

[1] Musser D. and Stepanov A., "Generic programming", in *International Symposium on Symbolic and Algebraic Computation*, 1988.

[2] Austern M. H., *Generic Programming and the STL: Using and Extending the C++ Standard Template Library*, ser. Addison-Wesley professional computing series. Addison-Wesley, 1999.

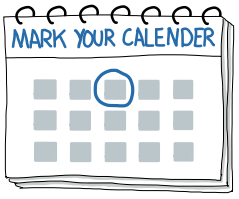
Images:
14: Franziska Panter
26: Franziska Panter

Upcoming Events

Training Classes

- *C++ Clean Code – Best Practices für Programmierer*, golem Akademie, September 13 - 17
- *Programmieren mit C++20*, Andreas Fertig, September 27 - 29
- *C++1x für eingebettete Systeme*, QA Systems, October 14 - 15

For my upcoming talks you can check https://andreasfertig.info/talks/.
For my courses you can check https://andreasfertig.info/courses/.
Like to always be informed? Subscribe to my newsletter: https://andreasfertig.info/newsletter/.



About Andreas Fertig

Photo: Kristijan Matic www.kristijanmatic.de

Andreas Fertig, CEO of Unique Code GmbH, is an experienced trainer and lecturer for C++ for standards 11 to 20.
Andreas is involved in the C++ standardization committee, in which the new standards are developed. At international conferences, he presents how code can be written better. He publishes specialist articles, e.g., for iX magazine, and has published several textbooks on C++.
With C++ Insights (https://cppinsights.io), Andreas has created an internationally recognized tool that enables users to look behind the scenes of C++ and thus to understand constructs even better.
Before working as a trainer and consultant, he worked for Philips Medizin Systeme GmbH for ten years as a C++ software developer and architect focusing on embedded systems.