Miscellaneous Improvements to C++11

Giovanni Dicanio
AUTHOR, SOFTWARE ENGINEER

https://blogs.msmvps.com/gdicanio

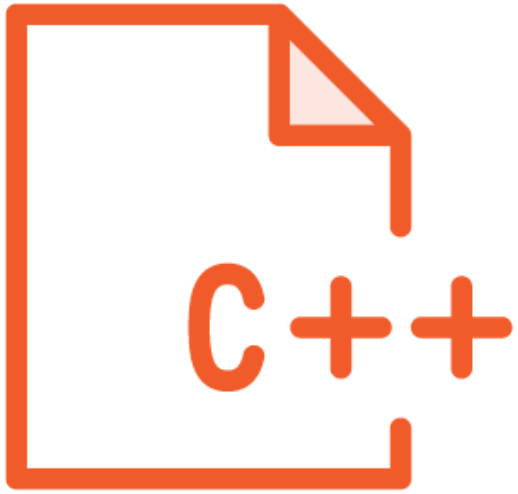
Overview

Relaxed constexpr functions

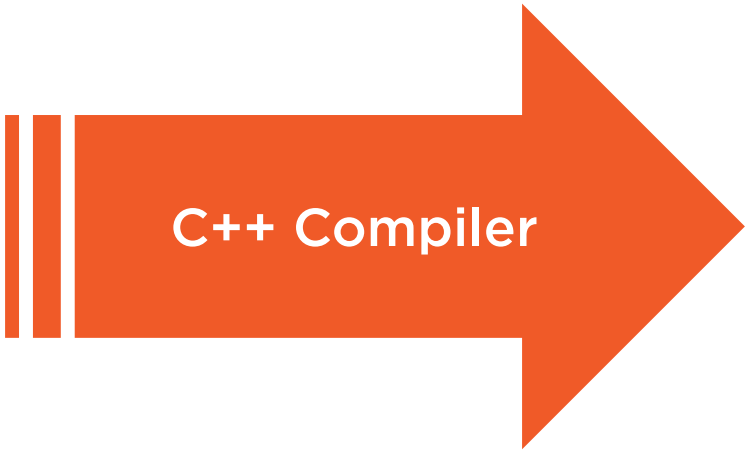
Variable templates

The [[deprecated]] attribute

The C++ Compilation Model

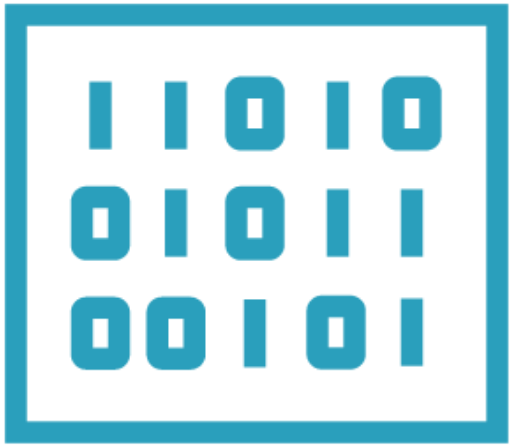


C++ Source Code



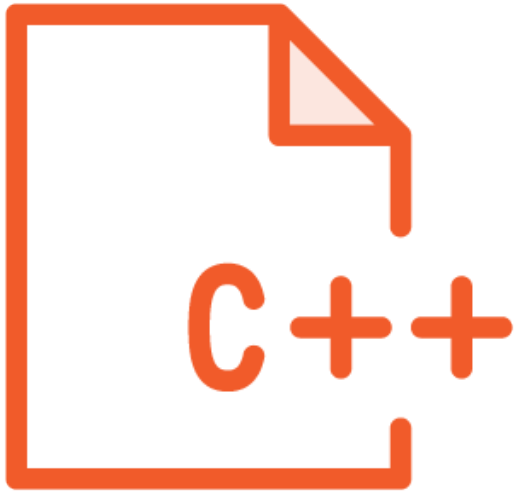
«C++11 from Scratch»
The C++ Compiler:
A Translator

bit.ly/CppComp

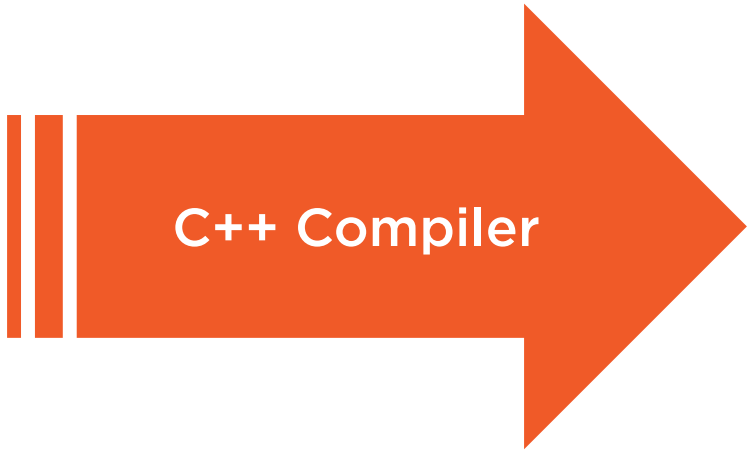


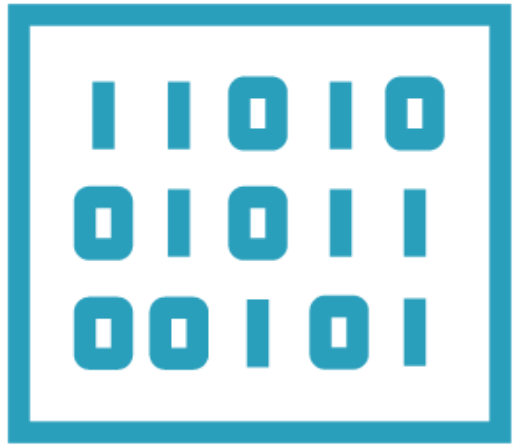
Executable Binary File

Run-time: The CPU Is Executing Your Program

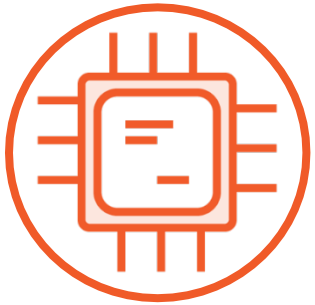


C++ Source Code



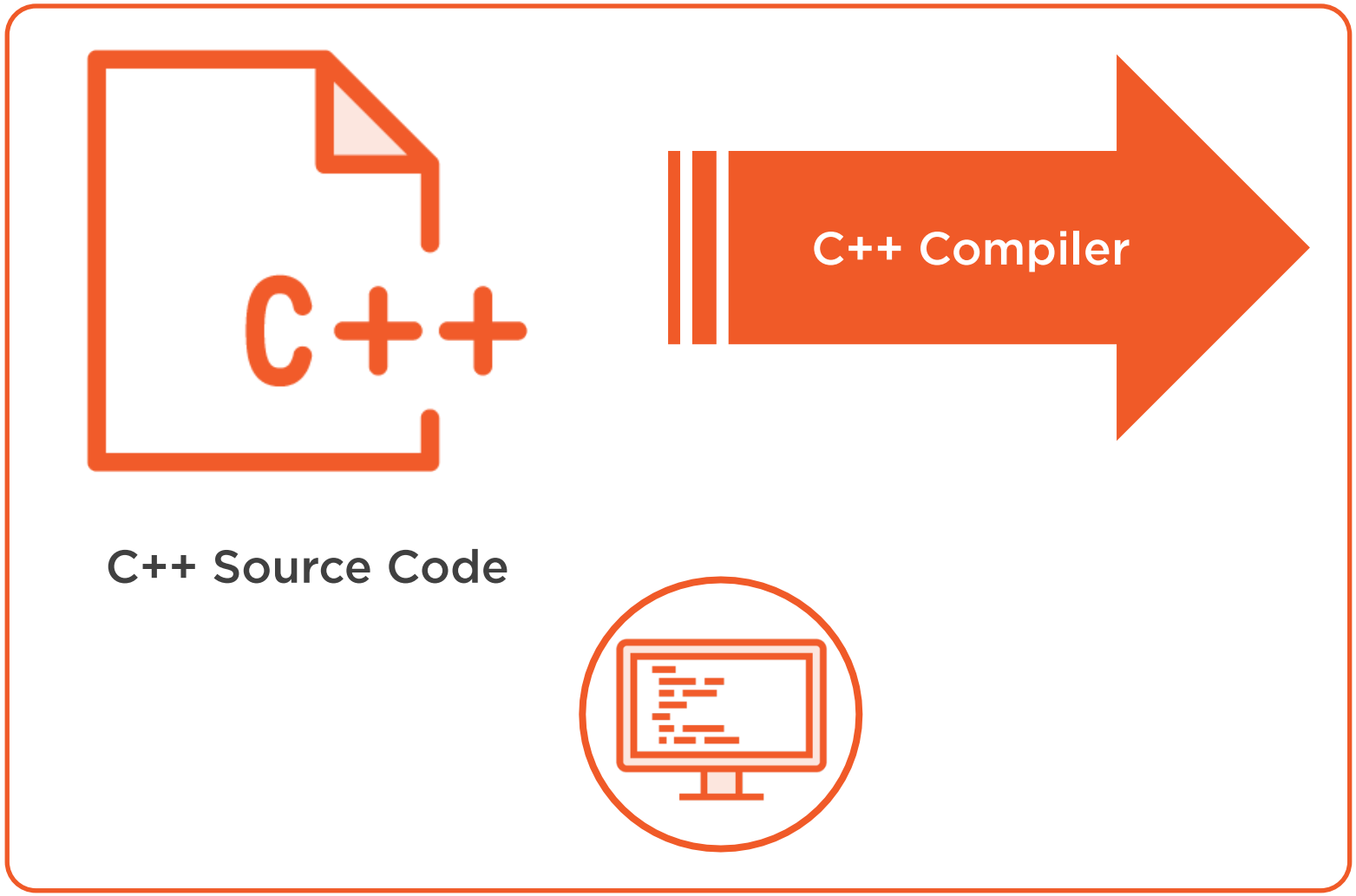


Executable Binary File



Run-time

Compile-time: The Compiler Is Translating Your Code



Terminal

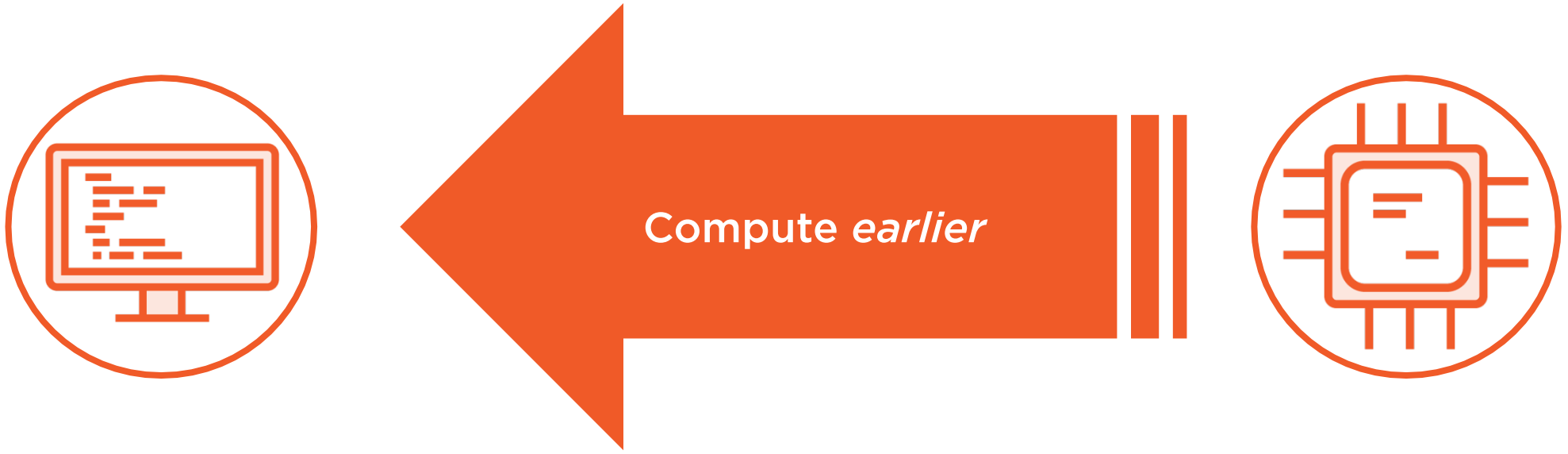
```cpp
#include <iostream>
using namespace std;

int main() {
  cin >> password;
  if (password == "Connie") {
    cout << "All right!\n";
  } else {
    cout << "Access denied.\n";
  }
}
```

giovanni@ubuntu: ~/projects

```
giovanni@ubuntu:~/projects$ clang++ Connie.cpp
Connie.cpp:5:10: error: use of undeclared identifier 'password'
  cin >> password;
         ^
Connie.cpp:6:7: error: use of undeclared identifier 'password'
  if (password == "Connie") {
      ^
2 errors generated.
giovanni@ubuntu:~/projects$
```

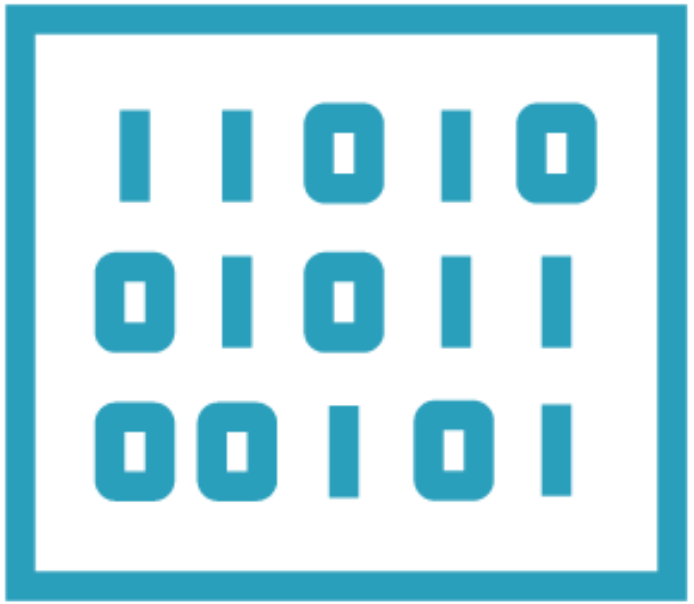
Compile-time

Compile-time Computation



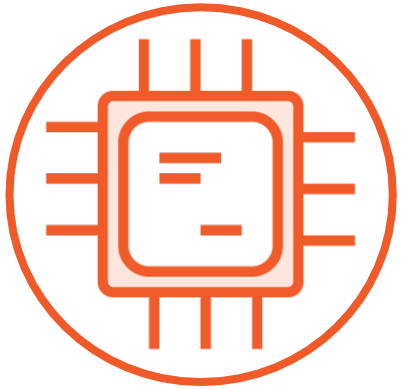
Compile-time Computation

Result *already* available at run-time



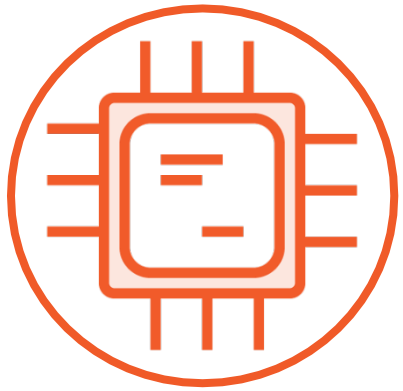
Executable Binary File

Just *look up* the result!



```
double DoSomething() {
...
}
```

Ordinary Functions Are Executed at Run-time



```
constexpr double DoSomething() {
    ...
}
```

constexpr Functions Executed at Compile-time

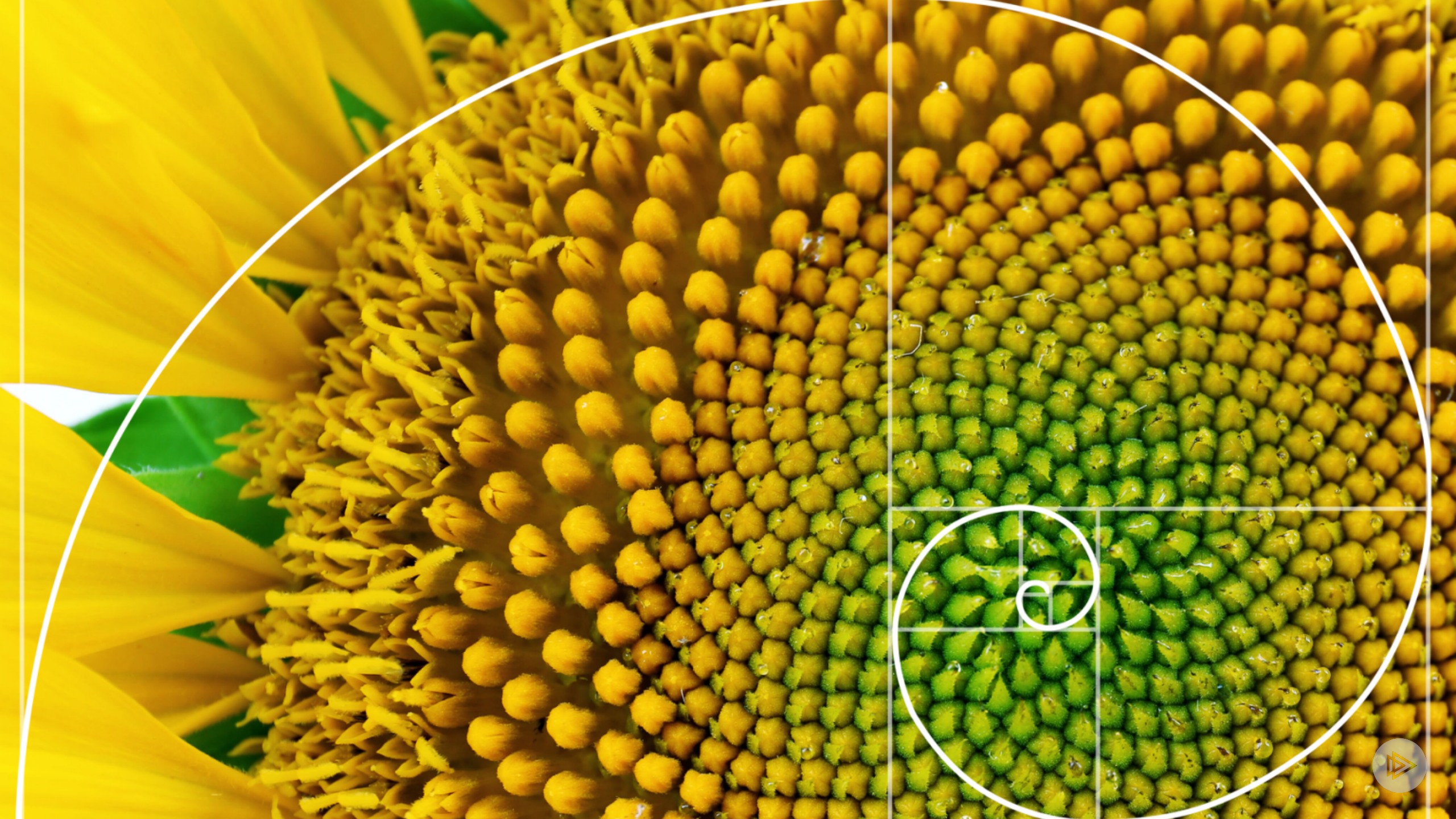
```
constexpr double DoSomething() {
    ...
}
```

C++14 *relaxed*
the highly-constraining C++11
constexpr requirements

constexpr Functions Executed at Compile-time

Demo

Compute Fibonacci numbers
at *compile-time*



Fibonacci Numbers

$$F_0 = 0$$

$$F_1 = 1$$

$$F_2 = F_1 + F_0$$

Add the *last two* numbers

Fibonacci Numbers

$$F_0 = 0$$

$$F_1 = 1$$

$$F_2 = F_1 + F_0 = 1 + 0 = 1$$

Fibonacci Numbers

$$F_0 = 0$$

$$F_1 = 1$$

$$F_2 = F_1 + F_0 = 1 + 0 = 1$$

$$F_3 = F_2 + F_1 = 1 + 1 = 2$$

Fibonacci Numbers

$$F_0 = 0$$

$$F_1 = 1$$

$$F_2 = F_1 + F_0 = 1 + 0 = 1$$

$$F_3 = F_2 + F_1 = 1 + 1 = 2$$

$$F_4 = F_3 + F_2 = 2 + 1 = 3$$

$$F_5 = F_4 + F_3 = 3 + 2 = 5$$

$$\dots$$

Class Templates

«C++11 from Scratch»

Meet std::vector: The Best Default Standard Container

bit.ly/CppStdVector

`vector<T>`

64	1980	77	500	2000

Class Templates

`vector<T>`

3.14	1.0	2.71	-3.2	9.12

Class Templates

vector<*T*>

Connie	John	Bill	Mike	C64

Class Templates

`vector<T>`

vector<*Image*>

Function Templates

```
template <typename T>
T Max(T a, T b) {
  if (a > b) {
    return a;
  }
  return b;
}
```

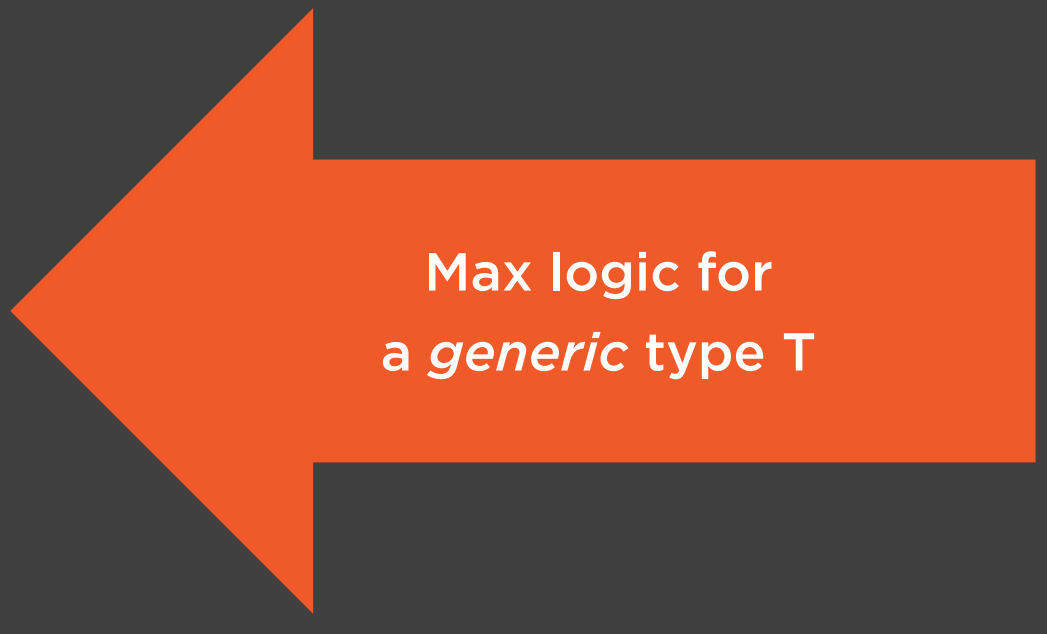
Function Templates

```
template <typename T>
T Max(T a, T b) {
  if (a > b) {
    return a;
  }
  return b;
}
```

Pass by const&
in production code

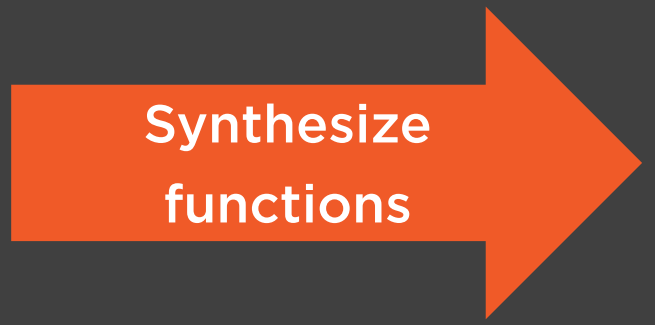
Function Templates

```
template <typename T>
T Max(T a, T b) {
  if (a > b) {
    return a;
  }
  return b;
}
```



Function Templates

```
template <typename T>
T Max(T a, T b) {
  if (a > b) {
    return a;
  }
  return b;
}
```




```
int Max(int, int)
double Max(double, double)
…
```

```
template <typename T>
constexpr T pi = T(3.14159265358979323846264338383);
```

C++14: Variable Templates

Value of pi for generic type T

pi<*double*>

```
template <typename T>
constexpr T pi = T(3.14159265358979323846264338383);
```

C++14: Variable Templates

Value of pi for generic type T

pi<*float*>

```
template <typename T>
constexpr T pi = T(3.14159265358979323846264338383);
```

C++14: Variable Templates

Value of pi for generic type T

Codebase Evolution

Initial Code

New Feature

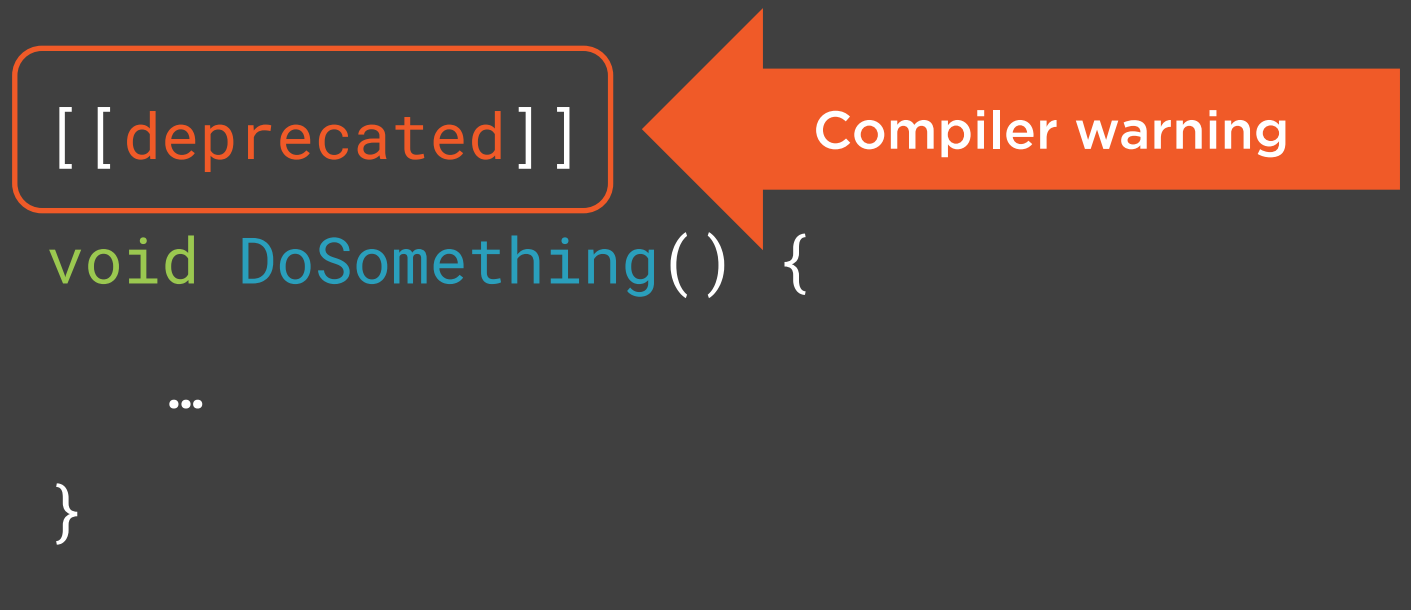
```
[[deprecated]]
void DoSomething() {
    …
}
```

The [[deprecated]] Attribute

```
[[deprecated]]
void DoSomething() {
    …
}
```

Please *don't* use!

The [[deprecated]] Attribute



The [[deprecated]] Attribute

Minimal deprecation syntax

Reason

```
[[deprecated("DoSomething is inefficient")]]
void DoSomething() {
    …
}
```

The [[deprecated]] Attribute

Using a custom message

Summary

(Relaxed) constexpr functions

Variable templates

[[deprecated]] attribute