

C with Classes  
oooooooo

Cfront era  
oooooooooooooooooooo

Standardization time  
oooooooooooo

C++ future  
oooo

(R)evolution!  
ooooo

Language popularity  
oooooooooooo

Summary  
oooooo

# (R)evolution of C++

## aka The Hitchhiker's Guide to C++

Łukasz Ziobroń

lukasz@ziobron.net

<http://ziobron.net>

code::dive, 2016-11-15

C with Classes  
oooooooo

Cfront era  
oooooooooooooooooooo

Standardization time  
oooooooooooo

C++ future  
oooo

(R)evolution!  
ooooo

Language popularity  
oooooooooooo

Summary  
oooooo

# About the author



C with Classes  
oooooooo

Cfront era  
oooooooooooooooooooo

Standardization time  
oooooooooooo

C++ future  
oooo

(R)evolution!  
ooooo

Language popularity  
oooooooooooo

Summary  
oooooo

# About the author



C with Classes  
oooooooo

Cfront era  
oooooooooooooooooooo

Standardization time  
oooooooooooo

C++ future  
oooo

(R)evolution!  
ooooo

Language popularity  
oooooooooooo

Summary  
oooooo

# About the author



C with Classes  
oooooooo

Cfront era  
oooooooooooooooooooo

Standardization time  
oooooooooooo

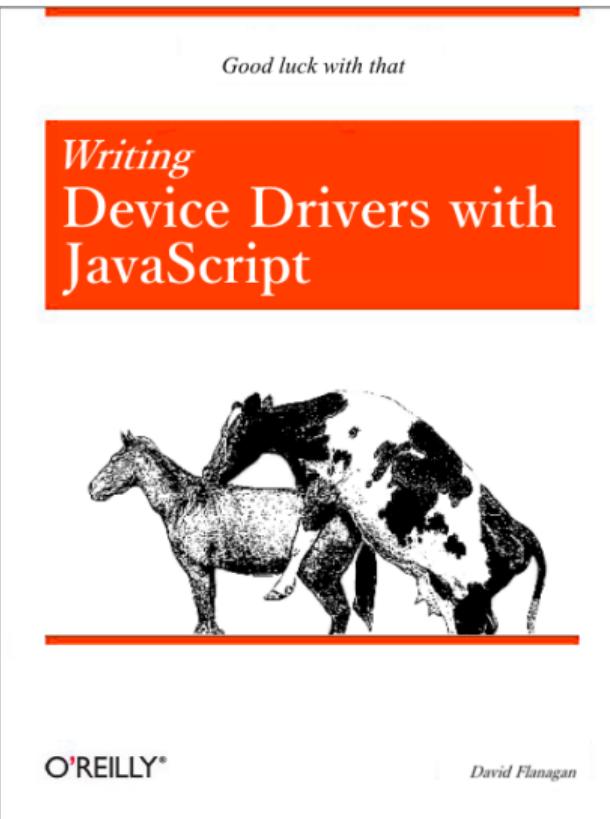
C++ future  
oooo

(R)evolution!  
oooo

Language popularity  
oooooooooooo

Summary  
oooooo

# (R)evolution of JS



C with Classes  
oooooooo

Cfront era  
oooooooooooooooooooo

Standardization time  
oooooooooooo

C++ future  
oooo

(R)evolution!  
ooooo

Language popularity  
oooooooooooo

Summary  
oooooo

# About the author



C with Classes  
oooooooo

Cfront era  
oooooooooooooooooooo

Standardization time  
oooooooooooo

C++ future  
oooo

(R)evolution!  
ooooo

Language popularity  
oooooooooooo

Summary  
oooooo

## About the author



C with Classes  
oooooooo

Cfront era  
oooooooooooooooooooo

Standardization time  
oooooooooooo

C++ future  
oooo

(R)evolution!  
ooooo

Language popularity  
oooooooooooo

Summary  
oooooo

## About the author



C with Classes  
oooooooo

Cfront era  
oooooooooooooooooooo

Standardization time  
oooooooooooo

C++ future  
oooo

(R)evolution!  
ooooo

Language popularity  
oooooooooooo

Summary  
oooooo

# About the author



C with Classes  
oooooooo

Cfront era  
oooooooooooooooooooo

Standardization time  
oooooooooooo

C++ future  
oooo

(R)evolution!  
ooooo

Language popularity  
oooooooooooo

Summary  
oooooo

# About the author



C with Classes  
oooooooo

Cfront era  
oooooooooooooooooooo

Standardization time  
oooooooooooo

C++ future  
oooo

(R)evolution!  
ooooo

Language popularity  
oooooooooooo

Summary  
oooooo

# About the author

## Interests:

- Archery
- Digital photography
- Machine learning
- Image processing
- High tech
- Starcraft
- Blogging at [ziobron.net](http://ziobron.net)



C with Classes  
oooooooo

Cfront era  
oooooooooooooooooooo

Standardization time  
oooooooooooo

C++ future  
oooo

(R)evolution!  
ooooo

Language popularity  
oooooooooooo

Summary  
oooooo

## Key messages

- ➊ C++ had a **clear aim**, which made it popular: to **organize code better without the loss of efficiency**

C with Classes  
oooooooo

Cfront era  
oooooooooooooooooooo

Standardization time  
oooooooooooo

C++ future  
oooo

(R)evolution!  
ooooo

Language popularity  
oooooooooooo

Summary  
oooooo

## Key messages

- ① C++ had a **clear aim**, which made it popular: to **organize code better without the loss of efficiency**
  
- ② C++ is even more popular now, because of new standards: **C++11** and **C++14**

C with Classes  
oooooooo

Cfront era  
oooooooooooooooooooo

Standardization time  
oooooooooooo

C++ future  
oooo

(R)evolution!  
ooooo

Language popularity  
oooooooooooo

Summary  
oooooo

## Key messages

- ① C++ had a **clear aim**, which made it popular: to **organize code better without the loss of efficiency**
- ② C++ is even more popular now, because of new standards: **C++11** and **C++14**
- ③ In future C++ will be **one of the most popular programming languages** so it's worth learning

C with Classes  
oooooooo

Cfront era  
oooooooooooooooooooo

Standardization time  
oooooooooooo

C++ future  
oooo

(R)evolution!  
ooooo

Language popularity  
oooooooooooo

Summary  
oooooo

# Agenda

- 1 C with Classes
- 2 Cfront era
- 3 Standardization time
- 4 C++ future
- 5 (R)evolution!
- 6 Language popularity
- 7 Summary

C with Classes  
oooooooo

Cfront era  
oooooooooooooooooooo

Standardization time  
oooooooooooo

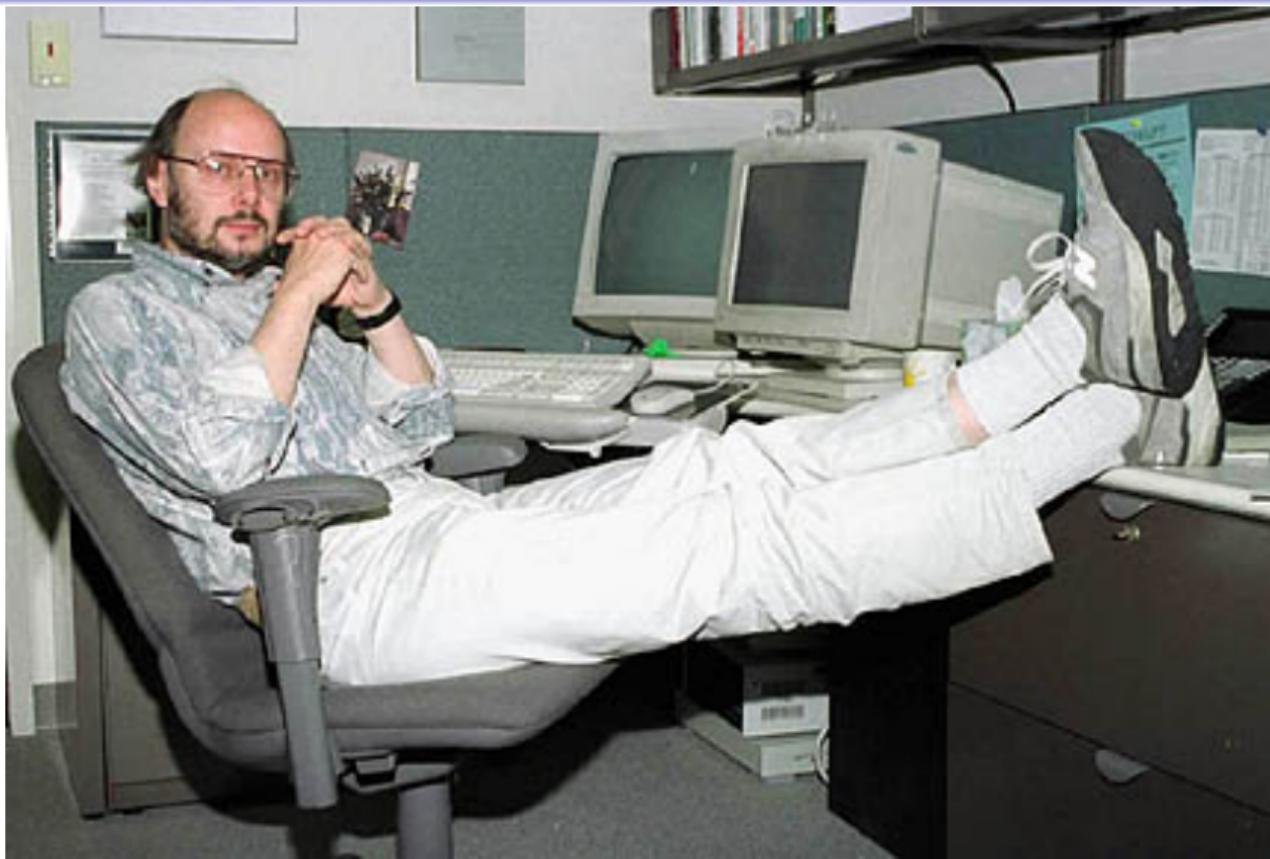
C++ future  
oooo

(R)evolution!  
ooooo

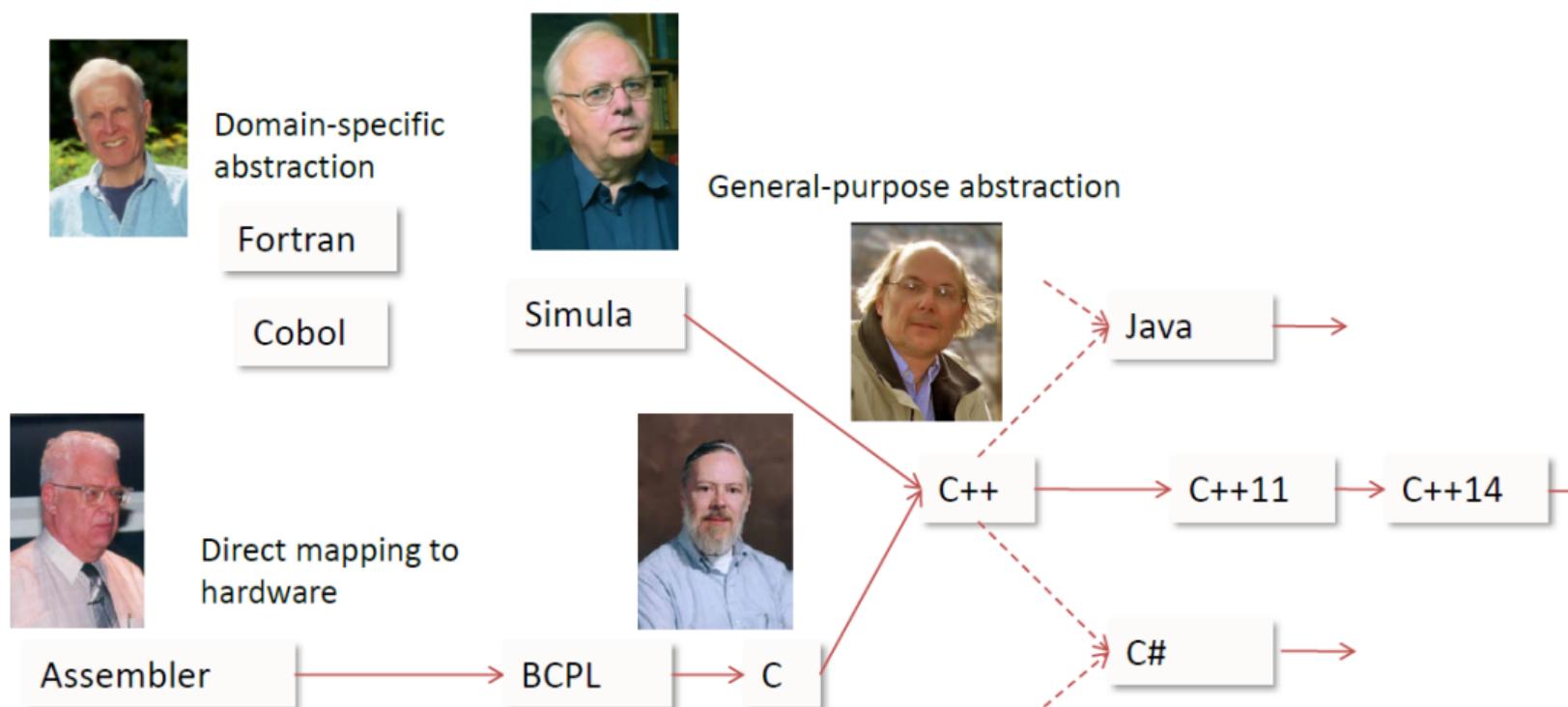
Language popularity  
oooooooooooo

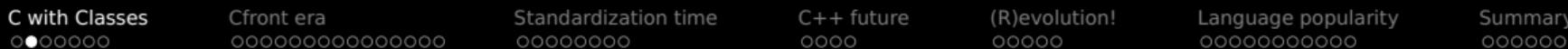
Summary  
oooooo

# C with Classes



## Roots of C++





## Roots of C++

Languages that were considered as a base of C++:

- Modula2
  - Ada
  - Smalltalk
  - Mesa
  - Clu
  - C

## C with Classes

Cfront era  
oooooooooooooooooooo

Standardization time  
oooooooo

## C++ future

(R)evolution!  
ooooo

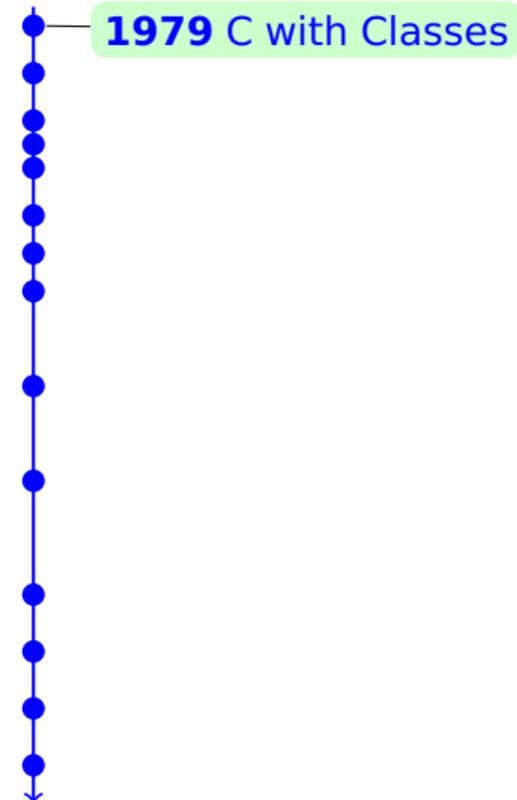
## Language popularity



# C with Classes

## Additions to C language:

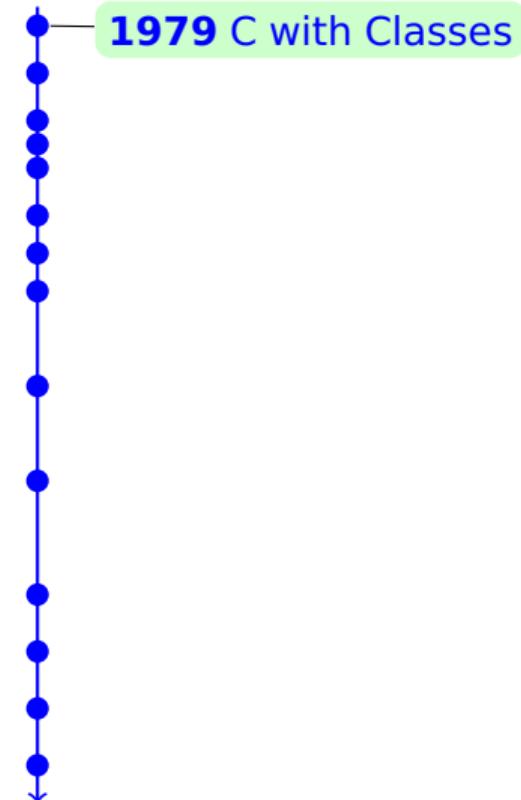
- classes



# C with Classes

## Additions to C language:

- classes
  - derived classes



## C with Classes

## Additions to C language:

- classes
  - derived classes
  - public and private access control

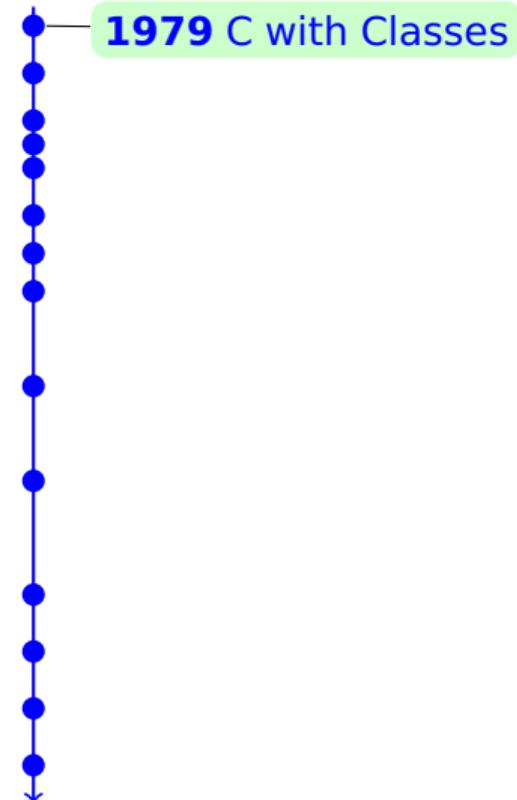
## 1979 C with Classes



## C with Classes

## Additions to C language:

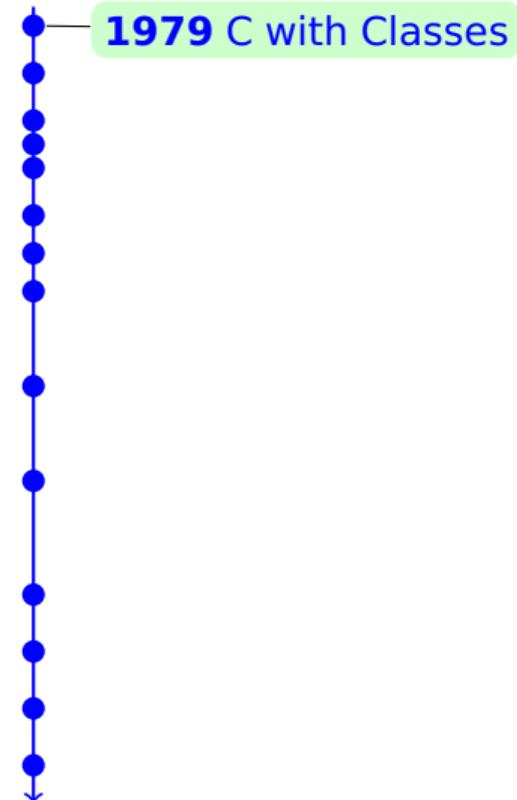
- classes
  - derived classes
  - public and private access control
  - constructors and destructors



# C with Classes

## Additions to C language:

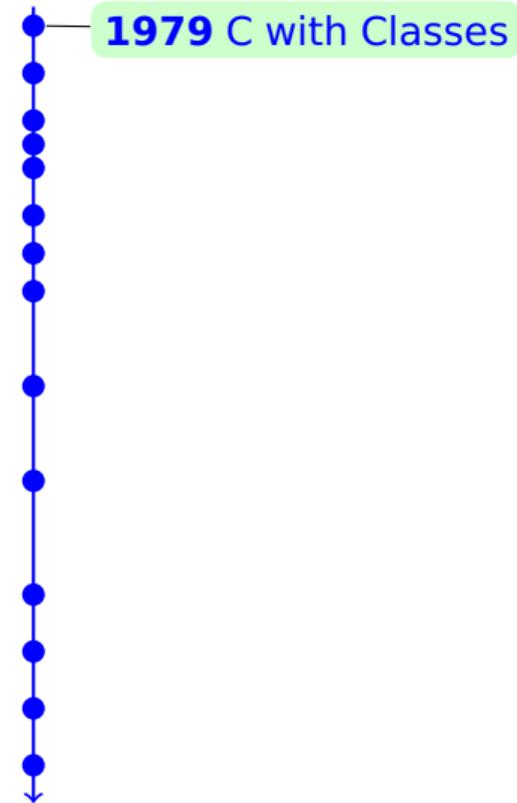
- classes
  - derived classes
  - public and private access control
  - constructors and destructors
  - call and return functions (removed later)



## C with Classes

## Additions to C language:

- classes
  - derived classes
  - public and private access control
  - constructors and destructors
  - call and return functions (removed later)
  - friend classes



# C with Classes

## Additions to C language:

- classes
  - derived classes
  - public and private access control
  - constructors and destructors
  - call and return functions (removed later)
  - friend classes
  - type checking and conversion of function arguments

## 1979 C with Classes

# C with Classes

## Additions to C language:

- classes
  - derived classes
  - public and private access control
  - constructors and destructors
  - **call and return functions (removed later)**
  - friend classes
  - type checking and conversion of function arguments

## 1979 C with Classes

# Example code in C with Classes

```
1 class stack {
2     char s[SIZE]; /* array of characters */
3     char * min;    /* pointer to bottom of stack */
4     char * top;    /* pointer to top of stack */
5     char * max;    /* pointer to top of allocated space */
6     void new();    /* initialization function (constructor) */
7     public:
8     void push(char);
9     char pop();
10 };
11 char stack.pop() // member functions always defined "elsewhere", not in class definition
12 {                // dot operator instead of double colon
13     if (top <= min) error("stack_underflow");
14     return * (--top);
15 }
16 class stack s1, s2; /* two variables of class stack, class was mandatory for declaration */
17 class stack * p1 = &s2; /* p1 points to s2 */
18 class stack * p2 = new stack; /* p2 points to stack object allocated on free store */
19 s1.push('h'); /* use object directly */
20 p1->push('s'); /* use object through pointer */
```

## Example code in C with Classes

```
1 class stack {
2     char s[SIZE]; /* array of characters */
3     char * min;    /* pointer to bottom of stack */
4     char * top;    /* pointer to top of stack */
5     char * max;    /* pointer to top of allocated space */
6     void new();    /* initialization function (constructor) */
7 public:
8     void push(char);
9     char pop();
10 };
11 char stack.pop() // member functions always defined "elsewhere", not in class definition
12 {               // dot operator instead of double colon
13     if (top <= min) error("stack_underflow");
14     return *(--top);
15 }
16 class stack s1, s2; /* two variables of class stack, class was mandatory for declaration */
17 class stack * p1 = &s2; /* p1 points to s2 */
18 class stack * p2 = new stack; /* p2 points to stack object allocated on free store */
19 s1.push('h'); /* use object directly */
20 p1->push('s'); /* use object through pointer */
```

## Example code in C with Classes

```
1 class stack {
2     char s[SIZE]; /* array of characters */
3     char * min;    /* pointer to bottom of stack */
4     char * top;    /* pointer to top of stack */
5     char * max;    /* pointer to top of allocated space */
6     void new();    /* initialization function (constructor) */
7 public:
8     void push(char);
9     char pop();
10};
11 char stack.pop() // member functions always defined "elsewhere", not in class definition
12 {               // dot operator instead of double colon
13     if (top <= min) error("stack_underflow");
14     return *(--top);
15}
16 class stack s1, s2; /* two variables of class stack, class was mandatory for declaration */
17 class stack * p1 = &s2; /* p1 points to s2 */
18 class stack * p2 = new stack; /* p2 points to stack object allocated on free store */
19 s1.push('h'); /* use object directly */
20 p1->push('s'); /* use object through pointer */
```

## Example code in C with Classes

```
1 class stack {
2     char s[SIZE]; /* array of characters */
3     char * min;    /* pointer to bottom of stack */
4     char * top;    /* pointer to top of stack */
5     char * max;    /* pointer to top of allocated space */
6     void new();    /* initialization function (constructor) */
7     public:
8     void push(char);
9     char pop();
10 };
11 char stack.pop() // member functions always defined "elsewhere", not in class definition
12 {               // dot operator instead of double colon
13     if (top <= min) error("stack_underflow");
14     return *(--top);
15 }
16 class stack s1, s2; /* two variables of class stack, class was mandatory for declaration */
17 class stack * p1 = &s2; /* p1 points to s2 */
18 class stack * p2 = new stack; /* p2 points to stack object allocated on free store */
19 s1.push('h'); /* use object directly */
20 p1->push('s'); /* use object through pointer */
```

## C with Classes

Cfront era  
oooooooooooooooo

Standardization time  
00000000

C++ future

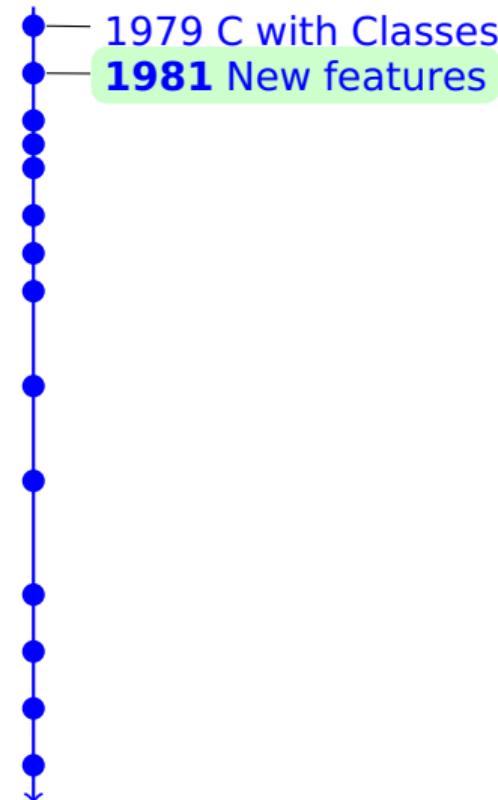
(R)evolution!  
99999

## Language popularity

# C with Classes

## New features added in 1981:

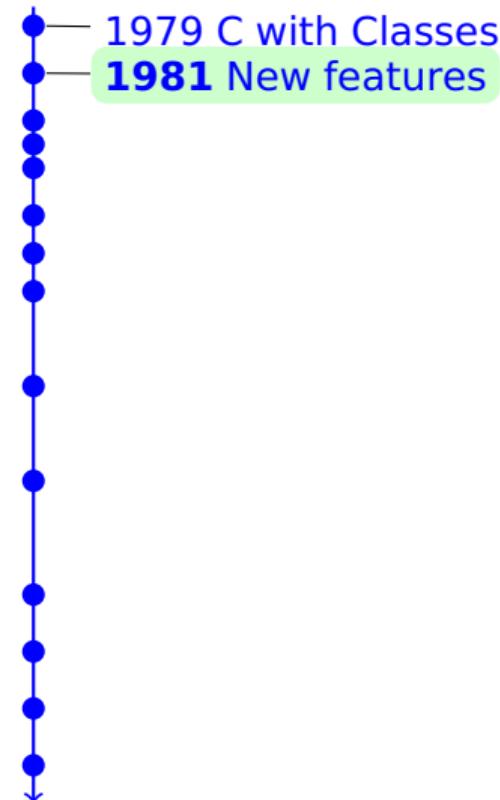
- inline functions



## C with Classes

#### New features added in 1981:

- inline functions
  - default arguments

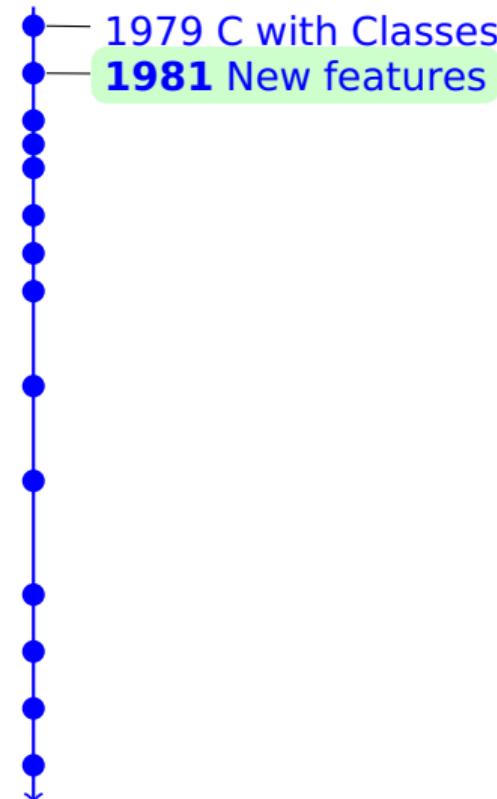




# C with Classes

## New features added in 1981:

- inline functions
  - default arguments
  - overloading of the assignment operator



## C with Classes

Cfront era  
oooooooooooooooo

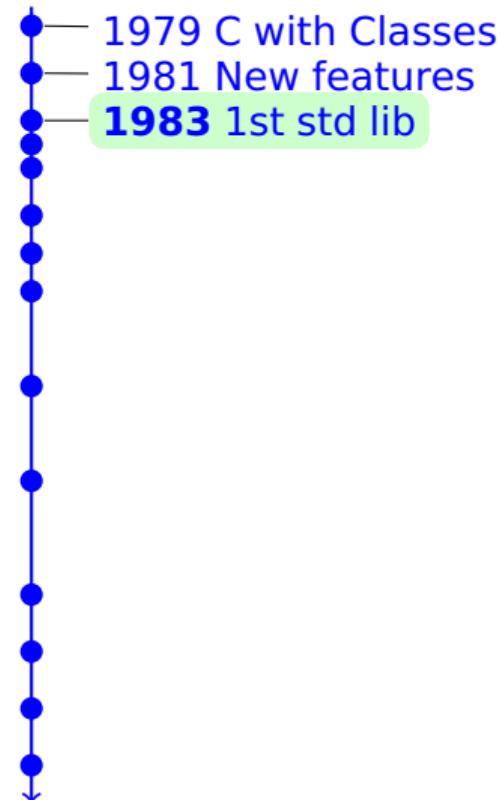
Standardization time  
00000000

## C++ future

(R)evolution!  
ooooo

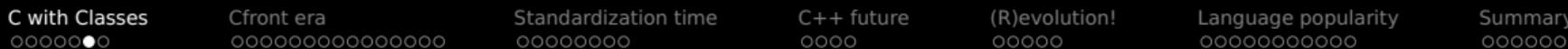
## Language popularity

## First standard library



## First elements in std lib:

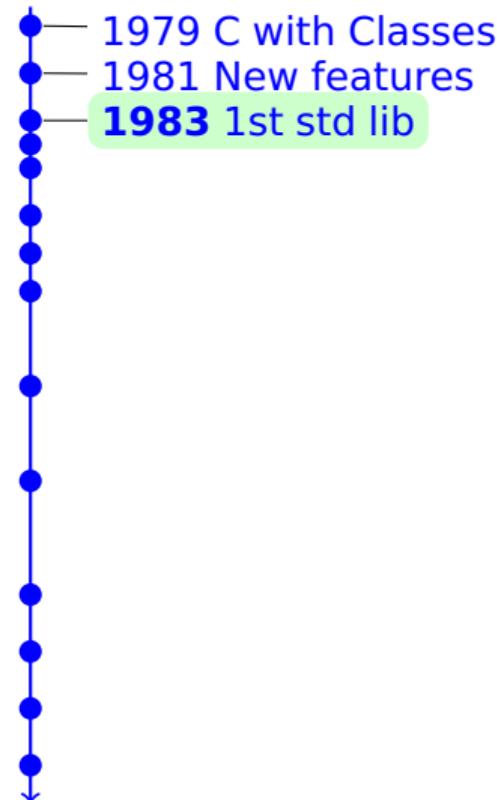
- complex numbers



## First standard library

## First elements in std lib:

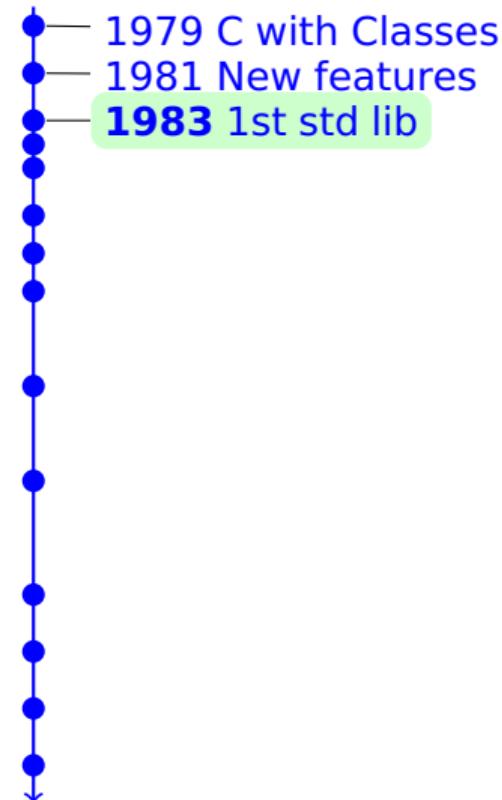
- complex numbers
  - string



## First standard library

## First elements in std lib:

- complex numbers
  - string
  - later: iostreams



C with Classes  
oooooooo●

Cfront era  
oooooooooooooooooooo

Standardization time  
oooooooooooo

C++ future  
oooo

(R)evolution!  
ooooo

Language popularity  
oooooooooooo

Summary  
ooooooo

# C with Classes - summary

- Years of development: 1979-1983

C with Classes  
oooooooo●

Cfront era  
oooooooooooooooooooo

Standardization time  
oooooooooooo

C++ future  
oooo

(R)evolution!  
ooooo

Language popularity  
oooooooooooo

Summary  
ooooooo

## C with Classes - summary

- Years of development: 1979-1983
- The idea was great

C with Classes  
oooooooo•

Cfront era  
oooooooooooooooooooo

Standardization time  
oooooooooooo

C++ future  
oooo

(R)evolution!  
ooooo

Language popularity  
oooooooooooo

Summary  
ooooooo

## C with Classes - summary

- Years of development: 1979-1983
- The idea was great
- The aim was clear:

C with Classes  
oooooooo•

Cfront era  
oooooooooooooooooooo

Standardization time  
oooooooooooo

C++ future  
oooo

(R)evolution!  
ooooo

Language popularity  
oooooooooooo

Summary  
oooooo

## C with Classes - summary

- Years of development: 1979-1983
- The idea was great
- The aim was clear:
  - help programmers to organize code with classes

C with Classes  
oooooooo•

Cfront era  
oooooooooooooooooooo

Standardization time  
oooooooooooo

C++ future  
oooo

(R)evolution!  
ooooo

Language popularity  
oooooooooooo

Summary  
oooooo

## C with Classes - summary

- Years of development: 1979-1983
- The idea was great
- The aim was clear:
  - help programmers to organize code with classes
  - without the loss of efficiency

C with Classes  
oooooooo•

Cfront era  
oooooooooooooooooooo

Standardization time  
oooooooooooo

C++ future  
oooo

(R)evolution!  
ooooo

Language popularity  
oooooooooooo

Summary  
oooooo

## C with Classes - summary

- Years of development: 1979-1983
- The idea was great
- The aim was clear:
  - help programmers to organize code with classes
  - without the loss of efficiency
  - and without requiring from users learning something completely new

C with Classes  
oooooooo•

Cfront era  
oooooooooooooooooooo

Standardization time  
oooooooooooo

C++ future  
oooo

(R)evolution!  
ooooo

Language popularity  
oooooooooooo

Summary  
oooooo

## C with Classes - summary

- Years of development: 1979-1983
- The idea was great
- The aim was clear:
  - help programmers to organize code with classes
  - without the loss of efficiency
  - and without requiring from users learning something completely new
- C with Classes didn't have many users :(

C with Classes  
oooooooo•

Cfront era  
oooooooooooooooooooo

Standardization time  
oooooooooooo

C++ future  
oooo

(R)evolution!  
ooooo

Language popularity  
oooooooooooo

Summary  
oooooo

## C with Classes - summary

- Years of development: 1979-1983
- The idea was great
- The aim was clear:
  - help programmers to organize code with classes
  - without the loss of efficiency
  - and without requiring from users learning something completely new
- C with Classes didn't have many users :(
- It wouldn't pay to support this language in the form it was

C with Classes  
oooooooo•

Cfront era  
oooooooooooooooooooo

Standardization time  
oooooooooooo

C++ future  
oooo

(R)evolution!  
ooooo

Language popularity  
oooooooooooo

Summary  
ooooooo

## C with Classes - summary

- Years of development: 1979-1983
- The idea was great
- The aim was clear:
  - help programmers to organize code with classes
  - without the loss of efficiency
  - and without requiring from users learning something completely new
- C with Classes didn't have many users :(
- It wouldn't pay to support this language in the form it was
- C with Classes was a "medium success"

C with Classes  
oooooooo•

Cfront era  
oooooooooooooooooooo

Standardization time  
oooooooooooo

C++ future  
oooo

(R)evolution!  
ooooo

Language popularity  
oooooooooooo

Summary  
oooooo

## C with Classes - summary

- Years of development: 1979-1983
- The idea was great
- The aim was clear:
  - help programmers to organize code with classes
  - without the loss of efficiency
  - and without requiring from users learning something completely new
- C with Classes didn't have many users :(
- It wouldn't pay to support this language in the form it was
- C with Classes was a "medium success"

C with Classes  
oooooooo•

Cfront era  
oooooooooooooooooooo

Standardization time  
oooooooooooo

C++ future  
oooo

(R)evolution!  
ooooo

Language popularity  
oooooooooooo

Summary  
oooooo

## C with Classes - summary

- Years of development: 1979-1983
- The idea was great
- The aim was clear:
  - help programmers to organize code with classes
  - without the loss of efficiency
  - and without requiring from users learning something completely new
- C with Classes didn't have many users :(
- It wouldn't pay to support this language in the form it was
- C with Classes was a "medium success"

Bjarne knew about it and drew conclusions.

C with Classes  
oooooooo

Cfront era  
oooooooooooooooooooo

Standardization time  
oooooooo

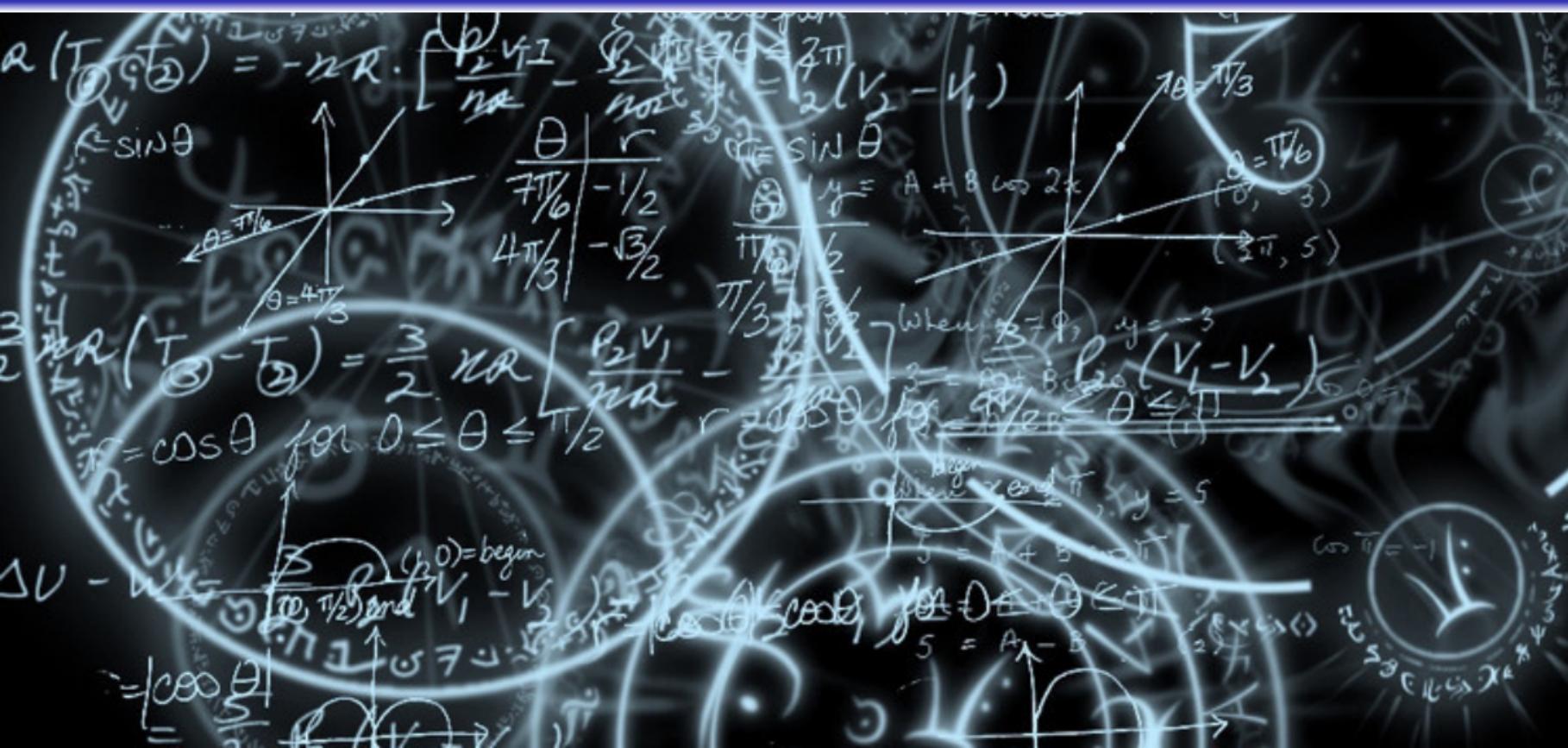
C++ future  
oooo

(R)evolution!  
oooo

Language popularity  
oooooooooooo

Summary  
oooooo

## Cfront era



C with Classes  
oooooooo

Cfront era  
●oooooooooooooo

Standardization time  
oooooooo

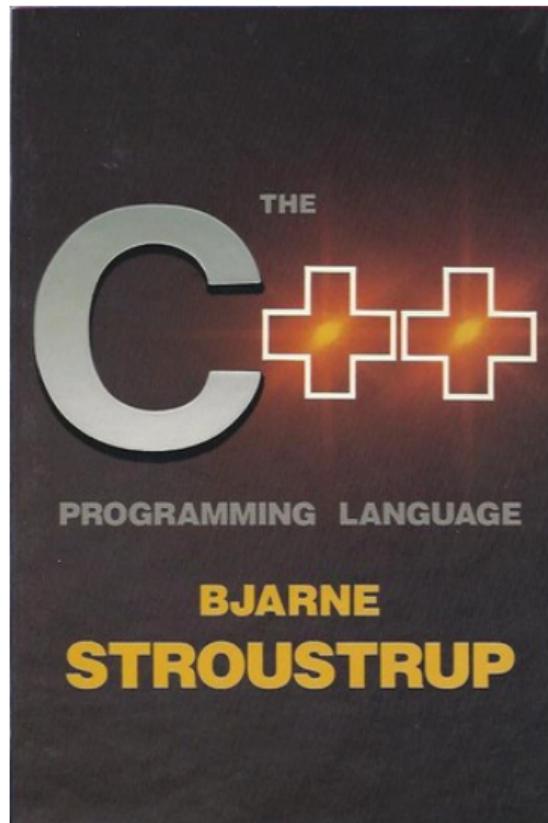
C++ future  
oooo

(R)evolution!  
ooooo

Language popularity  
oooooooooooo

Summary  
oooooo

# The C++ Programming language - 1st edition (1983)



C with Classes  
oooooooo

Cfront era  
●oooooooooooooo

Standardization time  
oooooooo

C++ future  
oooo

(R)evolution!  
oooo

Language popularity  
oooooooooooo

Summary  
oooooo

# The C++ Programming language - 1st edition (1983)

*How to actually learn any new programming concept*



*Essential*

Changing Stuff and  
Seeing What Happens

C with Classes  
oooooooo

Cfront era  
●oooooooooooooo

Standardization time  
oooooooo

C++ future  
oooo

(R)evolution!  
ooooo

Language popularity  
oooooooooooo

Summary  
oooooo

# The C++ Programming language - 1st edition (1983)

*Software can be chaotic, but we make it work*



*Expert*

Trying Stuff  
Until it Works

O RLY?

*The Practical Developer*  
@ThePracticalDev

C with Classes  
oooooooo

Cfront era  
○●oooooooooooooo

Standardization time  
oooooooo

C++ future  
oooo

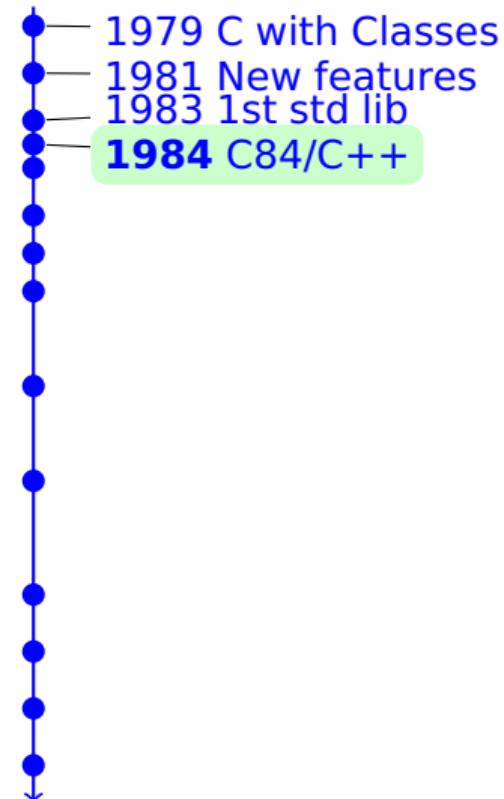
(R)evolution!  
ooooo

Language popularity  
oooooooooooo

Summary  
oooooo

## C84 and Cfront

- C with Classes got new name - C84



C with Classes  
oooooooo

Cfront era  
o●oooooooooooooo

Standardization time  
oooooooo

C++ future  
oooo

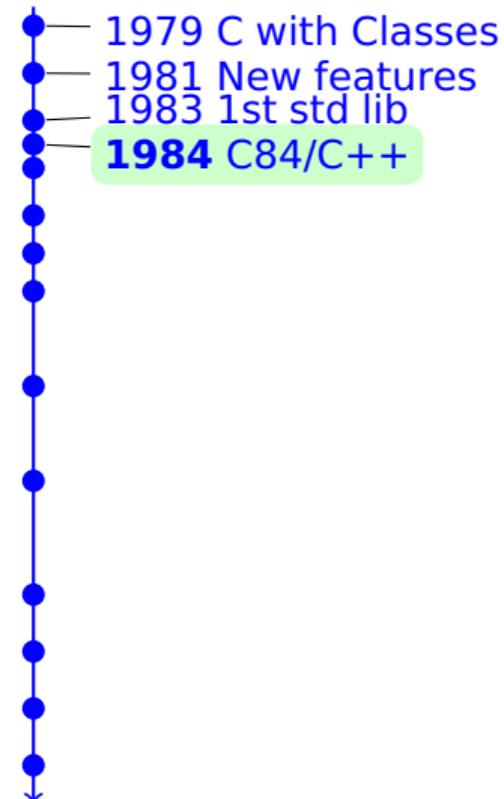
(R)evolution!  
ooooo

Language popularity  
oooooooooooo

Summary  
oooooo

## C84 and Cfront

- C with Classes got new name - C84
- A few months later C84 got a new name - C++



C with Classes  
oooooooo

Cfront era  
○●oooooooooooooo

Standardization time  
oooooooo

C++ future  
oooo

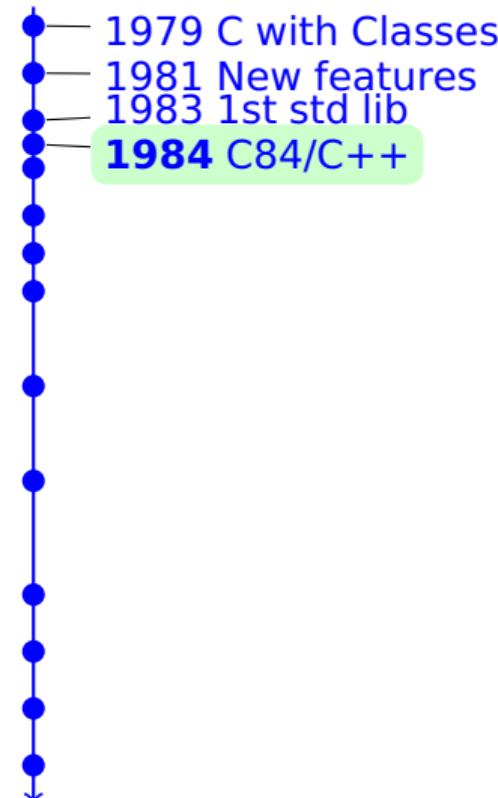
(R)evolution!  
ooooo

Language popularity  
oooooooooooo

Summary  
oooooo

## C84 and Cfront

- C with Classes got new name - C84
- A few months later C84 got a new name - C++
- First C++ compiler - Cfront



C with Classes  
oooooooo

Cfront era  
○●oooooooooooooo

Standardization time  
oooooooo

C++ future  
oooo

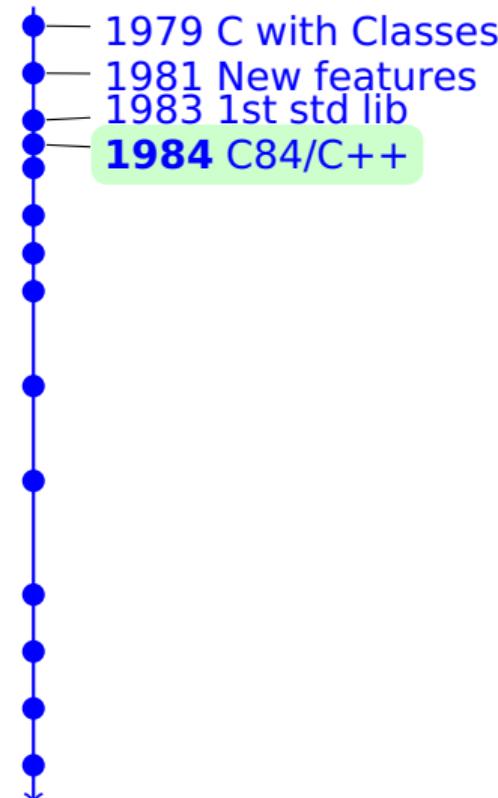
(R)evolution!  
ooooo

Language popularity  
oooooooooooo

Summary  
oooooo

## C84 and Cfront

- C with Classes got new name - C84
- A few months later C84 got a new name - C++
- First C++ compiler - Cfront
  - Originally written in... C with Classes



C with Classes  
oooooooo

Cfront era  
o●oooooooooooooo

Standardization time  
oooooooo

C++ future  
oooo

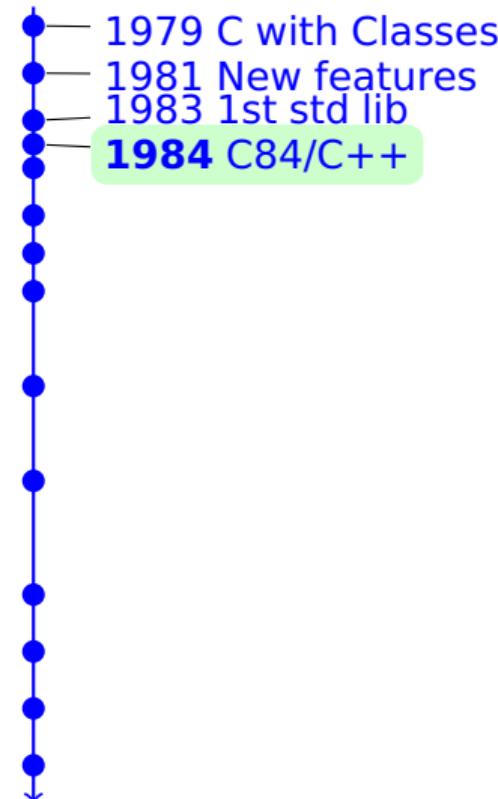
(R)evolution!  
ooooo

Language popularity  
oooooooooooo

Summary  
oooooo

## C84 and Cfront

- C with Classes got new name - C84
- A few months later C84 got a new name - C++
- First C++ compiler - Cfront
  - Originally written in... C with Classes
  - Transpiler to C code



C with Classes  
oooooooo

Cfront era  
○●oooooooooooooo

Standardization time  
oooooooo

C++ future  
oooo

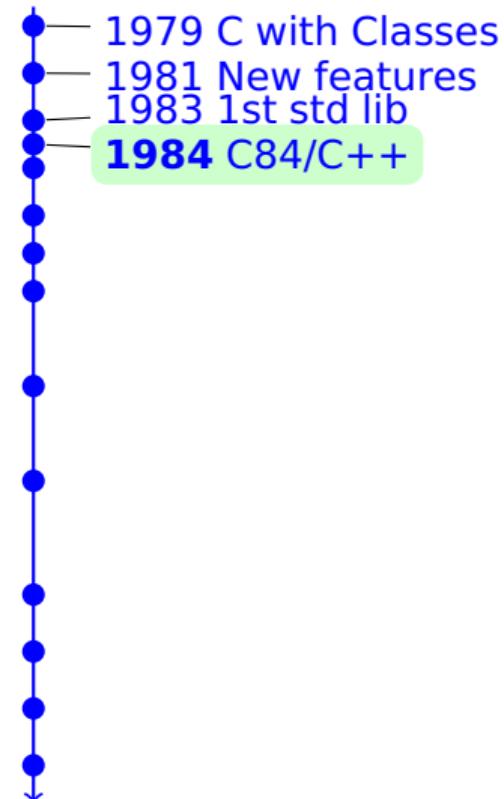
(R)evolution!  
ooooo

Language popularity  
oooooooooooo

Summary  
oooooo

## C84 and Cfront

- C with Classes got new name - C84
- A few months later C84 got a new name - C++
- First C++ compiler - Cfront
  - Originally written in... C with Classes
  - Transpiler to C code
  - Portability matters



C with Classes  
oooooooo

Cfront era  
○●oooooooooooooo

Standardization time  
oooooooo

C++ future  
oooo

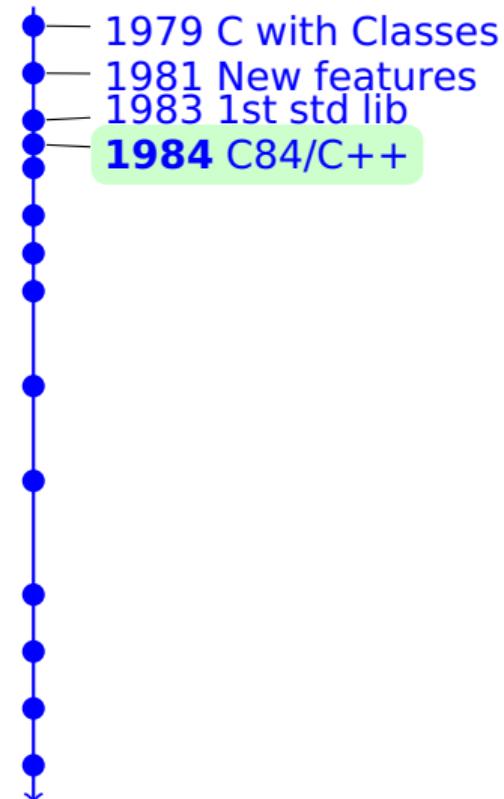
(R)evolution!  
ooooo

Language popularity  
oooooooooooo

Summary  
oooooo

## C84 and Cfront

- C with Classes got new name - C84
- A few months later C84 got a new name - C++
- First C++ compiler - Cfront
  - Originally written in... C with Classes
  - Transpiler to C code
  - Portability matters
  - C++ versions were named after Cfront releases



C with Classes  
oooooooo

Cfront era  
○○●oooooooooooo

Standardization time  
oooooooo

C++ future  
oooo

(R)evolution!  
ooooo

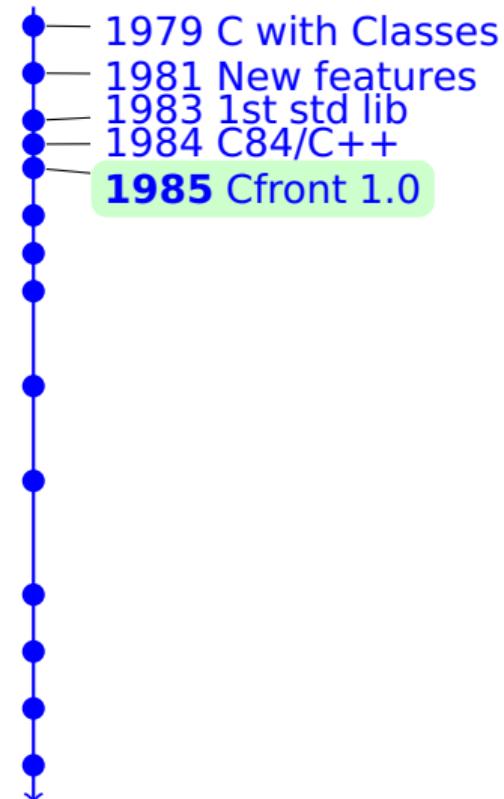
Language popularity  
oooooooooooo

Summary  
oooooo

# Cfront 1.0

New features:

- ➊ virtual functions



C with Classes  
oooooooo

Cfront era  
○○●oooooooooooo

Standardization time  
oooooooo

C++ future  
oooo

(R)evolution!  
ooooo

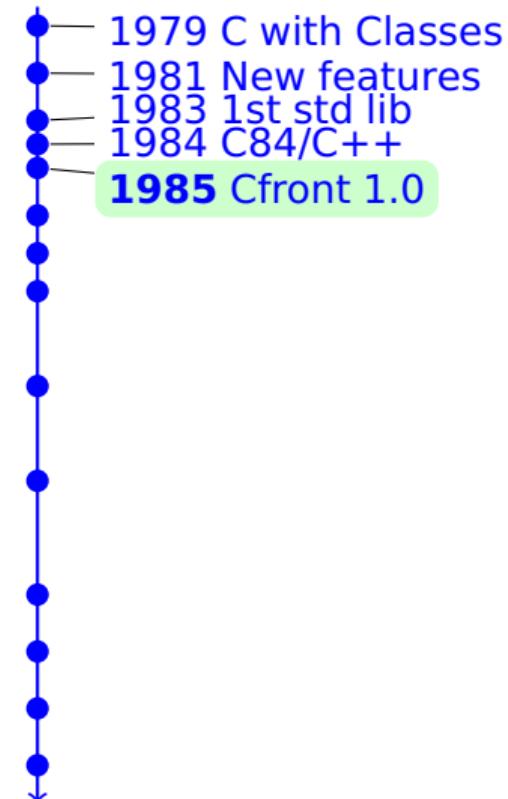
Language popularity  
oooooooooooo

Summary  
oooooo

# Cfront 1.0

New features:

- ① virtual functions
- ② function name and operator overloading



C with Classes  
oooooooo

Cfront era  
○○●oooooooooooo

Standardization time  
oooooooo

C++ future  
oooo

(R)evolution!  
oooo

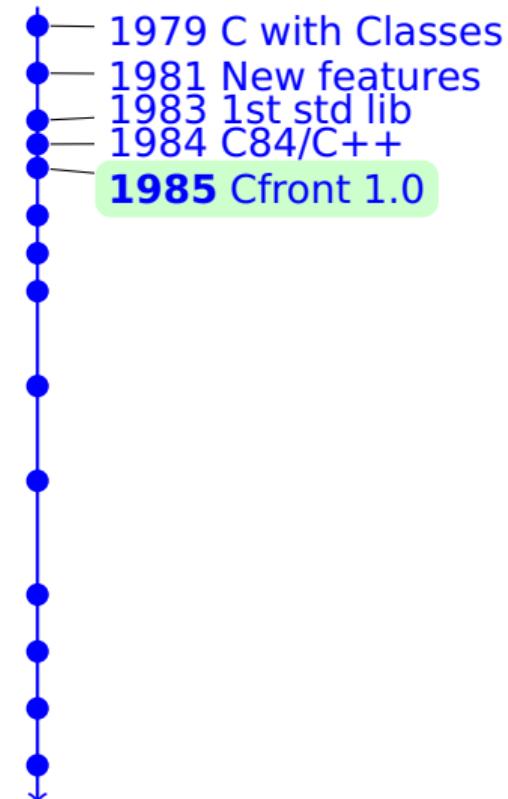
Language popularity  
oooooooo

Summary  
oooooo

# Cfront 1.0

New features:

- ① virtual functions
- ② function name and operator overloading
- ③ references



C with Classes  
oooooooo

Cfront era  
○○●oooooooooooo

Standardization time  
oooooooo

C++ future  
oooo

(R)evolution!  
oooo

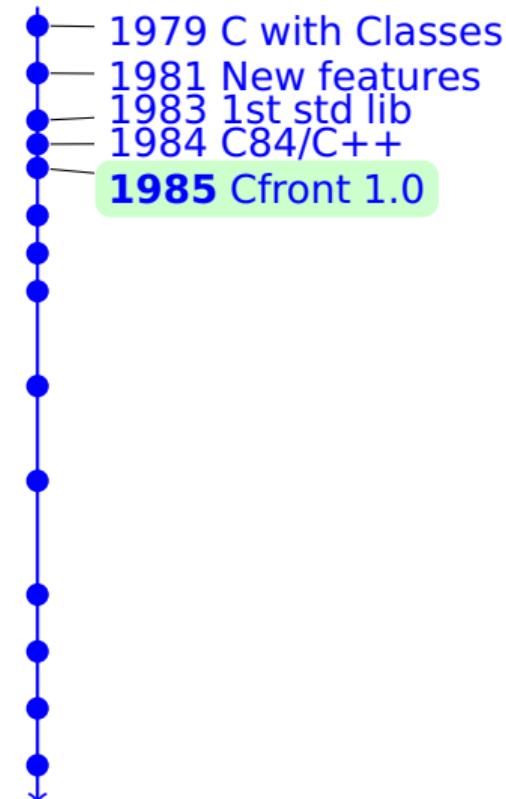
Language popularity  
oooooooo

Summary  
oooooo

# Cfront 1.0

New features:

- ① virtual functions
- ② function name and operator overloading
- ③ references
- ④ constants (const)



C with Classes  
oooooooo

Cfront era  
○○●oooooooooooo

Standardization time  
oooooooo

C++ future  
oooo

(R)evolution!  
oooo

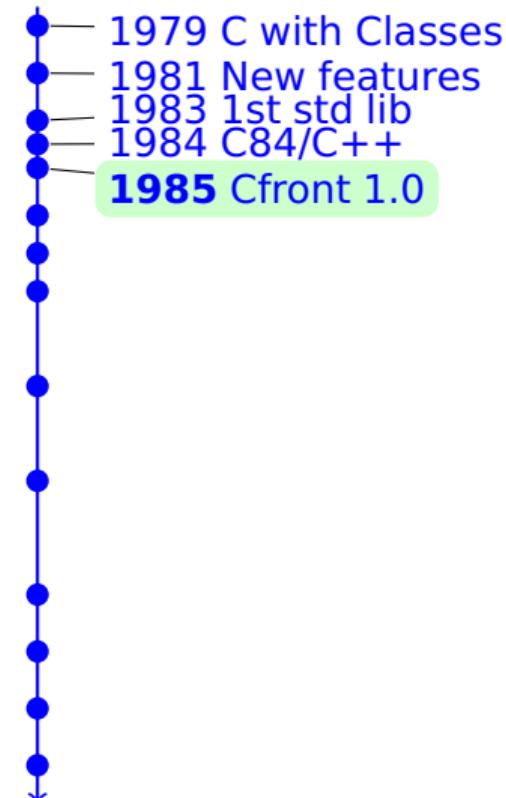
Language popularity  
oooooooo

Summary  
oooooo

# Cfront 1.0

New features:

- ① virtual functions
- ② function name and operator overloading
- ③ references
- ④ constants (const)
- ⑤ new and delete operators



C with Classes  
oooooooo

Cfront era  
○○●oooooooooooo

Standardization time  
oooooooo

C++ future  
oooo

(R)evolution!  
oooo

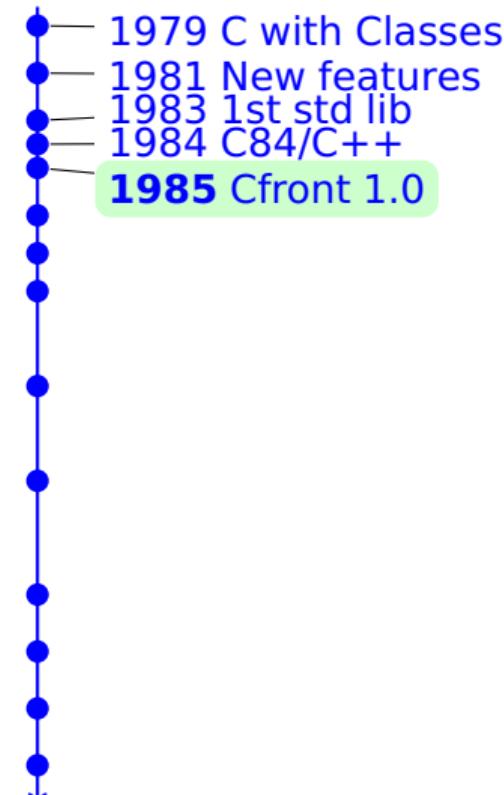
Language popularity  
oooooooo

Summary  
oooooo

# Cfront 1.0

New features:

- ① virtual functions
- ② function name and operator overloading
- ③ references
- ④ constants (const)
- ⑤ new and delete operators
- ⑥ improved type checking



C with Classes  
oooooooo

Cfront era  
○○●oooooooooooo

Standardization time  
oooooooo

C++ future  
oooo

(R)evolution!  
ooooo

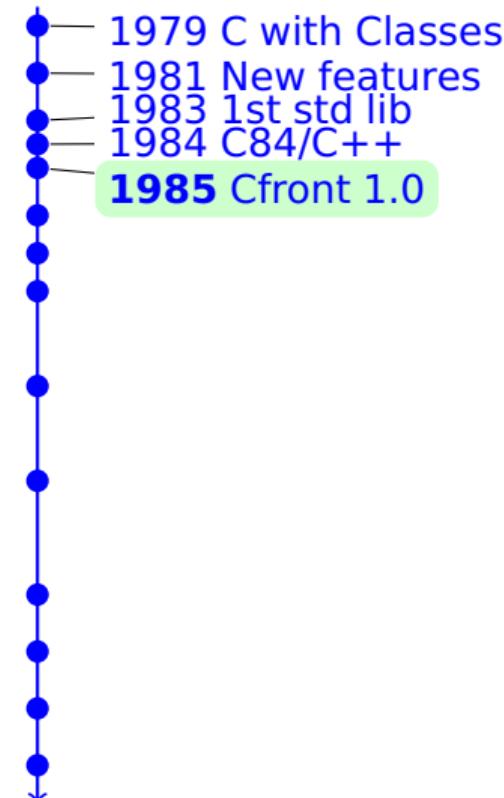
Language popularity  
oooooooooooo

Summary  
oooooo

# Cfront 1.0

New features:

- ① virtual functions
- ② function name and operator overloading
- ③ references
- ④ constants (const)
- ⑤ new and delete operators
- ⑥ improved type checking
- ⑦ scope resolution operator (::)



C with Classes  
oooooooo

Cfront era  
○○●oooooooooooo

Standardization time  
oooooooo

C++ future  
oooo

(R)evolution!  
oooo

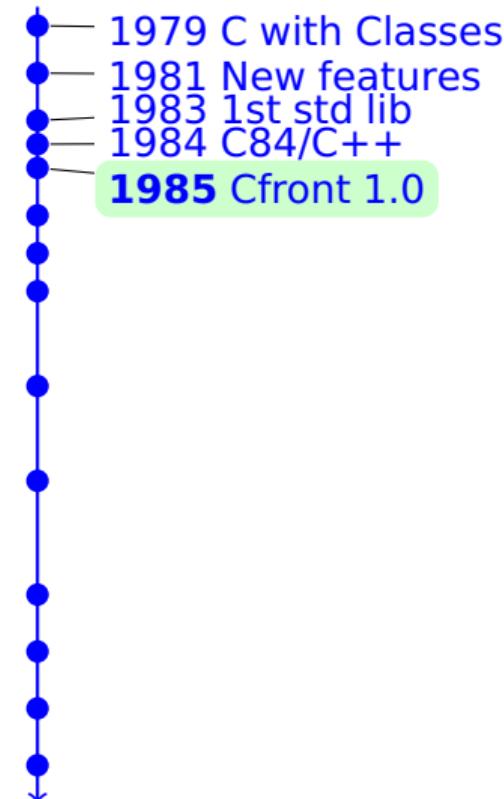
Language popularity  
oooooooo

Summary  
oooo

# Cfront 1.0

New features:

- ① virtual functions
- ② function name and operator overloading
- ③ references
- ④ constants (const)
- ⑤ new and delete operators
- ⑥ improved type checking
- ⑦ scope resolution operator (::)
- ⑧ BCPL-style comment terminated by end-of-line



C with Classes  
oooooooo

Cfront era  
oooo●oooooooooooo

Standardization time  
oooooooooooo

C++ future  
oooo

(R)evolution!  
ooooo

Language popularity  
oooooooooooo

Summary  
oooooo

# Example code in C++ 1.0

## Virtual functions:

```
1 class shape {  
2     point center;  
3     color col;  
4 public:  
5     point where() { return center; }  
6     void move(point to) { center = to; draw(); }  
7     virtual void draw();  
8     virtual void rotate(int);  
9 };  
10 class circle : public shape {  
11     int radius;  
12 public:  
13     void draw() { /* ... */ };  
14     void rotate(int) {} // yes, the null function  
15 };
```

C with Classes  
oooooooo

Cfront era  
oooo●oooooooooooo

Standardization time  
oooooooooooo

C++ future  
oooo

(R)evolution!  
ooooo

Language popularity  
oooooooooooo

Summary  
oooooo

# Example code in C++ 1.0

## Virtual functions:

```
1 class shape {  
2     point center;  
3     color col;  
4 public:  
5     point where() { return center; }  
6     void move(point to) { center = to; draw(); }  
7     virtual void draw();  
8     virtual void rotate(int);  
9 };  
10 class circle : public shape {  
11     int radius;  
12 public:  
13     void draw() { /* ... */ };  
14     void rotate(int) {} // yes, the null function  
15 };
```

C with Classes  
oooooooo

Cfront era  
oooo●oooooooooooo

Standardization time  
oooooooooooo

C++ future  
oooo

(R)evolution!  
ooooo

Language popularity  
oooooooooooo

Summary  
oooooo

# Example code in C++ 1.0

Overloaded functions:

```
1 overload print;  
2 void print(int);  
3 void print(const char*);
```

C with Classes  
oooooooo

Cfront era  
ooooo●oooooooooooo

Standardization time  
oooooooooooo

C++ future  
oooo

(R)evolution!  
ooooo

Language popularity  
oooooooooooo

Summary  
oooooo

# Example code in C++ 1.0

New and delete operators: Because you want to write that:

```
1 X* p = new X(2);
```

C with Classes  
oooooooo

Cfront era  
ooooo●oooooooooooo

Standardization time  
oooooooooooo

C++ future  
oooo

(R)evolution!  
ooooo

Language popularity  
oooooooooooo

Summary  
oooooo

## Example code in C++ 1.0

New and delete operators: Because you want to write that:

```
1 X* p = new X(2);
```

Instead of that:

```
1 struct X * p = (struct X *) malloc(sizeof(struct X));
2 if (p == 0) error("memory_exhausted");
3 p->init(2);
```

C with Classes  
oooooooo

Cfront era  
oooooooo●oooooooo

Standardization time  
oooooooo

C++ future  
oooo

(R)evolution!  
oooo

Language popularity  
oooooooooooo

Summary  
oooooo

## Example code in C++ 1.0

Improved type checking:

```
1 int printf(const char* ...); // accept any argument after
2 // the initial character string
3 printf("date:_%s_%d_19%d\n", month, day, year); // maybe right
```

C with Classes  
oooooooo

Cfront era  
oooooooo●oooooooo

Standardization time  
oooooooo

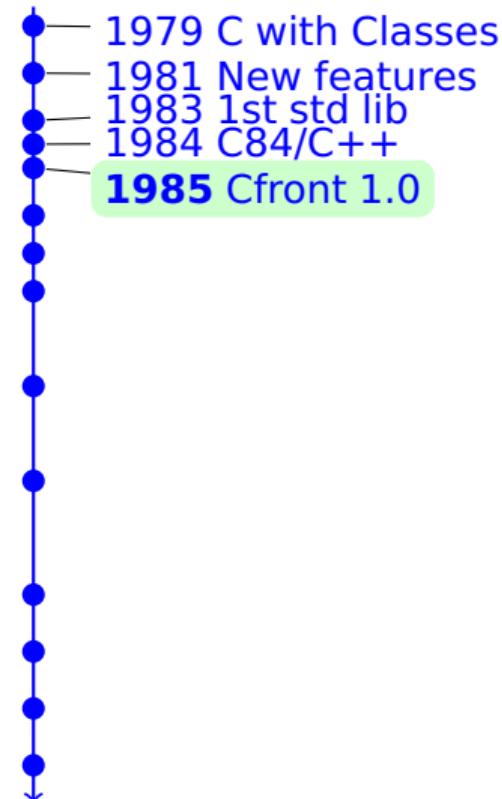
C++ future  
oooo

(R)evolution!  
oooo

Language popularity  
oooooooooooo

Summary  
oooooo

# Cfront 1.1 (1986) & Cfront 1.2 (1987)



New features:

- ① pointers to members
  - ② protected members
- + bug fixes

C with Classes  
oooooooo

Cfront era  
oooooooo●oooooooo

Standardization time  
oooooooo

C++ future  
oooo

(R)evolution!  
oooo

Language popularity  
oooooooooooo

Summary  
oooooo

# Cfront 2.0

New features:

C with Classes  
oooooooo

Cfront era  
oooooooo●oooooooo

Standardization time  
oooooooooooo

C++ future  
oooo

(R)evolution!  
ooooo

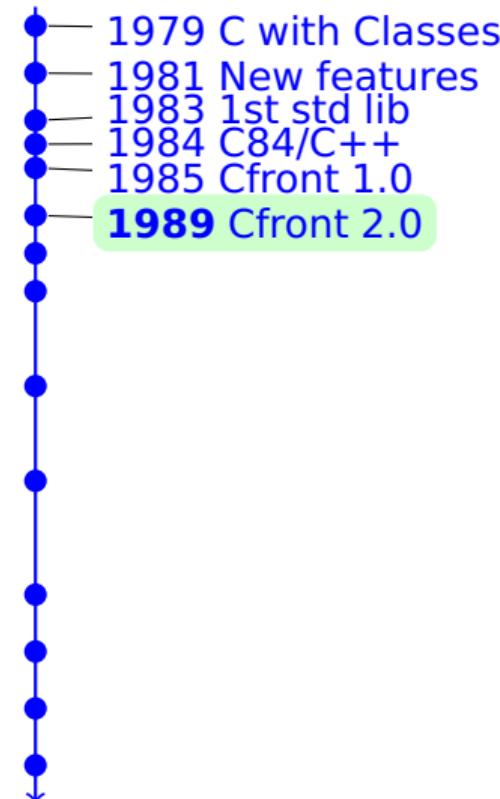
Language popularity  
oooooooooooo

Summary  
oooooo

# Cfront 2.0

New features:

- ① multiple inheritance
- ② type-safe linkage
- ③ recursive definition of assignment and initialization
- ④ abstract classes
- ⑤ static member functions
- ⑥ const member functions
- ⑦ overloading of operator ->



C with Classes  
oooooooo

Cfront era  
oooooooooooo●oooooo

Standardization time  
oooooooo

C++ future  
oooo

(R)evolution!  
ooooo

Language popularity  
oooooooooooo

Summary  
oooooo

# Cfront 3.0

New features:

C with Classes  
oooooooo

Cfront era  
oooooooooooo●ooooo

Standardization time  
oooooooo

C++ future  
oooo

(R)evolution!  
ooooo

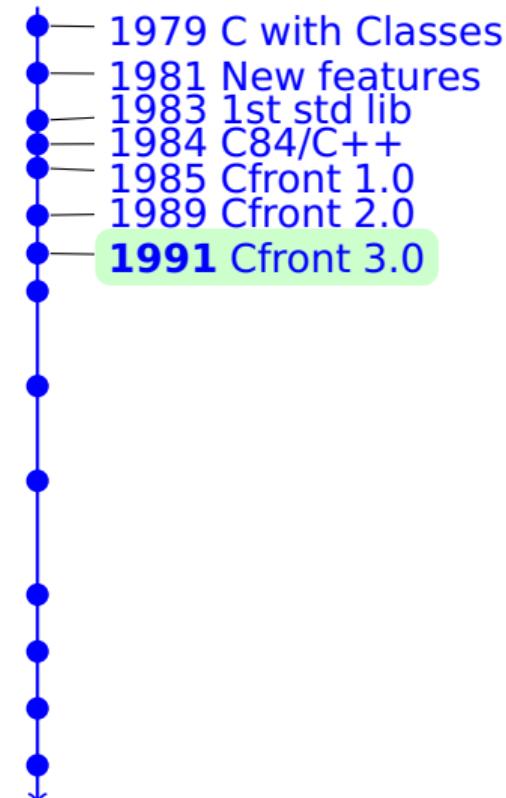
Language popularity  
oooooooooooo

Summary  
oooooo

# Cfront 3.0

New features:

- ① namespaces
- ② templates
- ③ nested classes
- ④ exceptions ("code approach")



C with Classes  
oooooooo

Cfront era  
oooooooooooo●oooo

Standardization time  
oooooooo

C++ future  
oooo

(R)evolution!  
oooo

Language popularity  
oooooooooooo

Summary  
oooooo

## Exceptions - code approach

- root of "bad exceptions" myth

---

<sup>1</sup>[http://www.baszerr.eu/lib/exe/fetch.php/docs/hello\\_houston.pdf](http://www.baszerr.eu/lib/exe/fetch.php/docs/hello_houston.pdf)

C with Classes  
oooooooo

Cfront era  
oooooooooooo●oooo

Standardization time  
oooooooo

C++ future  
oooo

(R)evolution!  
oooo

Language popularity  
oooooooooooo

Summary  
oooooo

## Exceptions - code approach

- root of "bad exceptions" myth
- runtime overhead

---

<sup>1</sup>[http://www.baszerr.eu/lib/exe/fetch.php/docs/hello\\_houston.pdf](http://www.baszerr.eu/lib/exe/fetch.php/docs/hello_houston.pdf)

C with Classes  
oooooooo

Cfront era  
oooooooooooo●oooo

Standardization time  
oooooooo

C++ future  
oooo

(R)evolution!  
oooo

Language popularity  
oooooooooooo

Summary  
oooooo

## Exceptions - code approach

- root of "bad exceptions" myth
- runtime overhead
- memory overhead

---

<sup>1</sup>[http://www.baszerr.eu/lib/exe/fetch.php/docs/hello\\_houston.pdf](http://www.baszerr.eu/lib/exe/fetch.php/docs/hello_houston.pdf)

C with Classes  
oooooooo

Cfront era  
oooooooooooo●oooo

Standardization time  
oooooooo

C++ future  
oooo

(R)evolution!  
oooo

Language popularity  
oooooooooooo

Summary  
oooooo

## Exceptions - code approach

- root of "bad exceptions" myth
- runtime overhead
- memory overhead
- legends about resource consumption



---

<sup>1</sup>[http://www.baszerr.eu/lib/exe/fetch.php/docs/hello\\_houston.pdf](http://www.baszerr.eu/lib/exe/fetch.php/docs/hello_houston.pdf)

# Exceptions - code approach

- root of "bad exceptions" myth
- runtime overhead
- memory overhead
- legends about resource consumption
- see: Bartosz 'BaSz' Szurgot - *Hello Huston*<sup>1</sup>



---

<sup>1</sup>[http://www.baszerr.eu/lib/exe/fetch.php/docs/hello\\_houston.pdf](http://www.baszerr.eu/lib/exe/fetch.php/docs/hello_houston.pdf)

C with Classes  
oooooooo

Cfront era  
oooooooooooo●ooo

Standardization time  
oooooooo

C++ future  
oooo

(R)evolution!  
oooo

Language popularity  
oooooooooooo

Summary  
oooooo

## Exceptions - table approach

- no runtime overhead (if no exception)

---

<sup>2</sup>[http://www.baszerr.eu/lib/exe/fetch.php/docs/hello\\_houston.pdf](http://www.baszerr.eu/lib/exe/fetch.php/docs/hello_houston.pdf)

C with Classes  
oooooooo

Cfront era  
oooooooooooo●ooo

Standardization time  
oooooooo

C++ future  
oooo

(R)evolution!  
oooo

Language popularity  
oooooooooooo

Summary  
oooooo

## Exceptions - table approach

- no runtime overhead (if no exception)
- no memory overhead

---

<sup>2</sup>[http://www.baszerr.eu/lib/exe/fetch.php/docs/hello\\_houston.pdf](http://www.baszerr.eu/lib/exe/fetch.php/docs/hello_houston.pdf)

C with Classes  
oooooooo

Cfront era  
oooooooooooo●ooo

Standardization time  
oooooooo

C++ future  
oooo

(R)evolution!  
oooo

Language popularity  
oooooooooooo

Summary  
oooooo

## Exceptions - table approach

- no runtime overhead (if no exception)
- no memory overhead
- increased binary size

---

<sup>2</sup>[http://www.baszerr.eu/lib/exe/fetch.php/docs/hello\\_houston.pdf](http://www.baszerr.eu/lib/exe/fetch.php/docs/hello_houston.pdf)

C with Classes  
oooooooo

Cfront era  
oooooooooooo●ooo

Standardization time  
oooooooo

C++ future  
oooo

(R)evolution!  
oooo

Language popularity  
oooooooooooo

Summary  
oooooo

## Exceptions - table approach

- no runtime overhead (if no exception)
- no memory overhead
- increased binary size
- unknown exception handling time

---

<sup>2</sup>[http://www.baszerr.eu/lib/exe/fetch.php/docs/hello\\_houston.pdf](http://www.baszerr.eu/lib/exe/fetch.php/docs/hello_houston.pdf)

C with Classes  
oooooooo

Cfront era  
oooooooooooo●ooo

Standardization time  
oooooooo

C++ future  
oooo

(R)evolution!  
oooo

Language popularity  
oooooooooooo

Summary  
oooooo

## Exceptions - table approach

- no runtime overhead (if no exception)
- no memory overhead
- increased binary size
- unknown exception handling time
- the fastest solution (if no exception)

---

<sup>2</sup>[http://www.baszerr.eu/lib/exe/fetch.php/docs/hello\\_houston.pdf](http://www.baszerr.eu/lib/exe/fetch.php/docs/hello_houston.pdf)

C with Classes  
oooooooo

Cfront era  
oooooooooooo●ooo

Standardization time  
oooooooo

C++ future  
oooo

(R)evolution!  
oooo

Language popularity  
oooooooooooo

Summary  
oooooo

## Exceptions - table approach

- no runtime overhead (if no exception)
- no memory overhead
- increased binary size
- unknown exception handling time
- the fastest solution (if no exception)
- impossible without compiler support

---

<sup>2</sup>[http://www.baszerr.eu/lib/exe/fetch.php/docs/hello\\_houston.pdf](http://www.baszerr.eu/lib/exe/fetch.php/docs/hello_houston.pdf)

# Exceptions - table approach

- no runtime overhead (if no exception)
- no memory overhead
- increased binary size
- unknown exception handling time
- the fastest solution (if no exception)
- impossible without compiler support
- exceptions vs Cfront



<sup>2</sup>[http://www.baszerr.eu/lib/exe/fetch.php/docs/hello\\_houston.pdf](http://www.baszerr.eu/lib/exe/fetch.php/docs/hello_houston.pdf)

# Exceptions - table approach

- no runtime overhead (if no exception)
- no memory overhead
- increased binary size
- unknown exception handling time
- the fastest solution (if no exception)
- impossible without compiler support
- exceptions vs Cfront
- see: Bartosz 'BaSz' Szurgot - *Hello Huston*<sup>2</sup>



<sup>2</sup>[http://www.baszerr.eu/lib/exe/fetch.php/docs/hello\\_houston.pdf](http://www.baszerr.eu/lib/exe/fetch.php/docs/hello_houston.pdf)

C with Classes  
oooooooo

Cfront era  
oooooooooooo●oo

Standardization time  
oooooooo

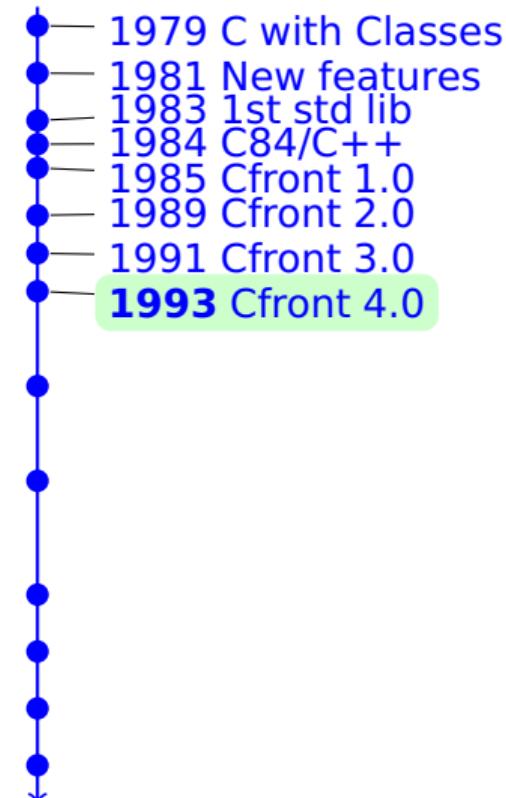
C++ future  
oooo

(R)evolution!  
ooooo

Language popularity  
oooooooooooo

Summary  
oooooo

# Cfront 4.0



New features:

- ① exceptions ("table approach")

C with Classes  
oooooooo

Cfront era  
oooooooooooooo●○

Standardization time  
oooooooo

C++ future  
oooo

(R)evolution!  
oooo

Language popularity  
oooooooooooo

Summary  
oooooo

# Cfront RIP



C with Classes  
oooooooo

Cfront era  
oooooooooooooo●

Standardization time  
oooooooo

C++ future  
oooo

(R)evolution!  
ooooo

Language popularity  
oooooooooooo

Summary  
oooooo

# Cfront - summary

- Years of development: 1985-1993

C with Classes  
oooooooo

Cfront era  
oooooooooooooo●

Standardization time  
oooooooo

C++ future  
oooo

(R)evolution!  
ooooo

Language popularity  
oooooooooooo

Summary  
oooooo

# Cfront - summary

- Years of development: 1985-1993
- Compiler's frontend

C with Classes  
oooooooo

Cfront era  
oooooooooooooo●

Standardization time  
oooooooo

C++ future  
oooo

(R)evolution!  
ooooo

Language popularity  
oooooooooooo

Summary  
oooooo

## Cfront - summary

- Years of development: 1985-1993
- Compiler's frontend
- Translated C++ to C (free portability)

C with Classes  
oooooooo

Cfront era  
oooooooooooooo●

Standardization time  
oooooooo

C++ future  
oooo

(R)evolution!  
ooooo

Language popularity  
oooooooooooo

Summary  
oooooo

## Cfront - summary

- Years of development: 1985-1993
- Compiler's frontend
- Translated C++ to C (free portability)
- Explosion of interest:

C with Classes  
oooooooo

Cfront era  
oooooooooooooo●

Standardization time  
oooooooo

C++ future  
oooo

(R)evolution!  
ooooo

Language popularity  
oooooooooooo

Summary  
oooooo

## Cfront - summary

- Years of development: 1985-1993
- Compiler's frontend
- Translated C++ to C (free portability)
- Explosion of interest:
  - commercial release in 1985

C with Classes  
oooooooo

Cfront era  
oooooooooooooo●

Standardization time  
oooooooo

C++ future  
oooo

(R)evolution!  
ooooo

Language popularity  
oooooooooooo

Summary  
oooooo

## Cfront - summary

- Years of development: 1985-1993
- Compiler's frontend
- Translated C++ to C (free portability)
- Explosion of interest:
  - commercial release in 1985
  - from **500 users in 1985...**

C with Classes  
oooooooo

Cfront era  
oooooooooooooo●

Standardization time  
oooooooo

C++ future  
oooo

(R)evolution!  
ooooo

Language popularity  
oooooooooooo

Summary  
oooooo

## Cfront - summary

- Years of development: 1985-1993
- Compiler's frontend
- Translated C++ to C (free portability)
- Explosion of interest:
  - commercial release in 1985
  - from **500 users in 1985...**
  - to **over 1 500 000 users in 1993!**

C with Classes  
oooooooo

Cfront era  
oooooooooooooo●

Standardization time  
oooooooo

C++ future  
oooo

(R)evolution!  
ooooo

Language popularity  
oooooooooooo

Summary  
oooooo

## Cfront - summary

- Years of development: 1985-1993
- Compiler's frontend
- Translated C++ to C (free portability)
- Explosion of interest:
  - commercial release in 1985
  - from **500 users in 1985...**
  - to **over 1 500 000 users in 1993!**
  - exponential growth!

C with Classes  
oooooooo

Cfront era  
oooooooooooooo●

Standardization time  
oooooooo

C++ future  
oooo

(R)evolution!  
ooooo

Language popularity  
oooooooooooo

Summary  
oooooo

## Cfront - summary

- Years of development: 1985-1993
- Compiler's frontend
- Translated C++ to C (free portability)
- Explosion of interest:
  - commercial release in 1985
  - from **500 users in 1985...**
  - to **over 1 500 000 users in 1993!**
  - exponential growth!
- Many new features added

C with Classes  
oooooooo

Cfront era  
oooooooooooooo●

Standardization time  
oooooooo

C++ future  
oooo

(R)evolution!  
oooo

Language popularity  
oooooooooooo

Summary  
oooooo

## Cfront - summary

- Years of development: 1985-1993
- Compiler's frontend
- Translated C++ to C (free portability)
- Explosion of interest:
  - commercial release in 1985
  - from **500 users in 1985...**
  - to **over 1 500 000 users in 1993!**
  - exponential growth!
- Many new features added
- New approach to exceptions was impossible to add

C with Classes  
oooooooo

Cfront era  
oooooooooooooooooooo

Standardization time  
oooooooooooo

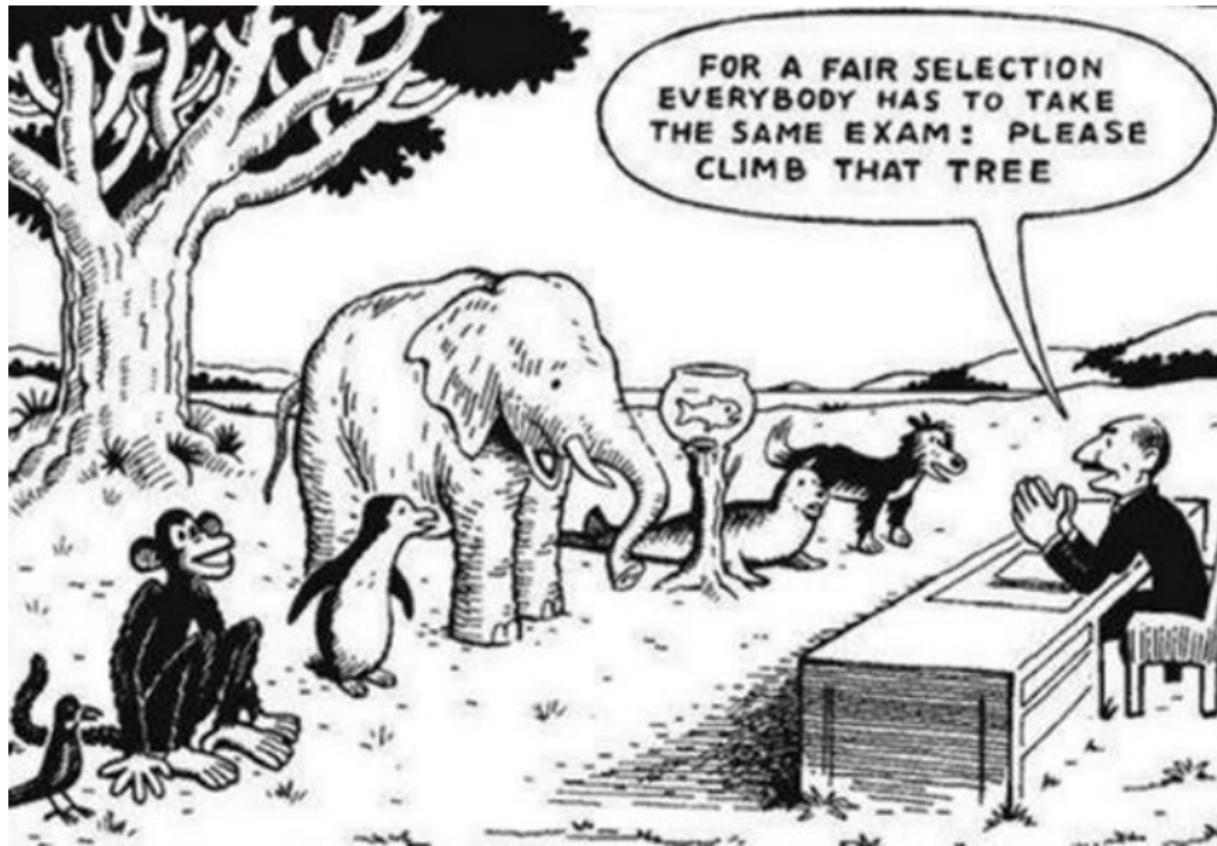
C++ future  
oooo

(R)evolution!  
oooo

Language popularity  
oooooooooooo

Summary  
oooooo

## Standardization time



C with Classes  
oooooooo

Cfront era  
oooooooooooooooooooo

Standardization time  
●oooooooooooo

C++ future  
oooo

(R)evolution!  
ooooo

Language popularity  
oooooooooooo

Summary  
ooooooo

# 1998 - First ISO C++ standard

Features:

C with Classes  
oooooooo

Cfront era  
oooooooooooooooooooo

Standardization time  
●oooooooo

C++ future  
oooo

(R)evolution!  
ooooo

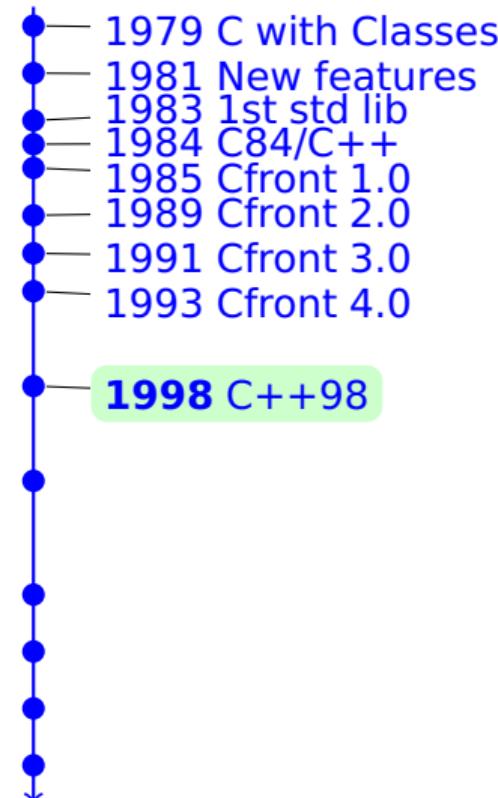
Language popularity  
oooooooooooo

Summary  
oooooo

# 1998 - First ISO C++ standard

## Features:

- RTTI (dynamic\_cast, typeid)
- cast operators
- mutable
- bool
- declarations in conditions
- template instantiations
- member templates
- export



C with Classes  
oooooooo

Cfront era  
oooooooooooooooooooo

Standardization time  
○●oooooooo

C++ future  
oooo

(R)evolution!  
ooooo

Language popularity  
oooooooooooo

Summary  
oooooo

# 1998 - First ISO C++ standard

Library additions:

C with Classes  
oooooooo

Cfront era  
oooooooooooooooooooo

Standardization time  
○●○○○○○○

C++ future  
○○○○

(R)evolution!  
○○○○

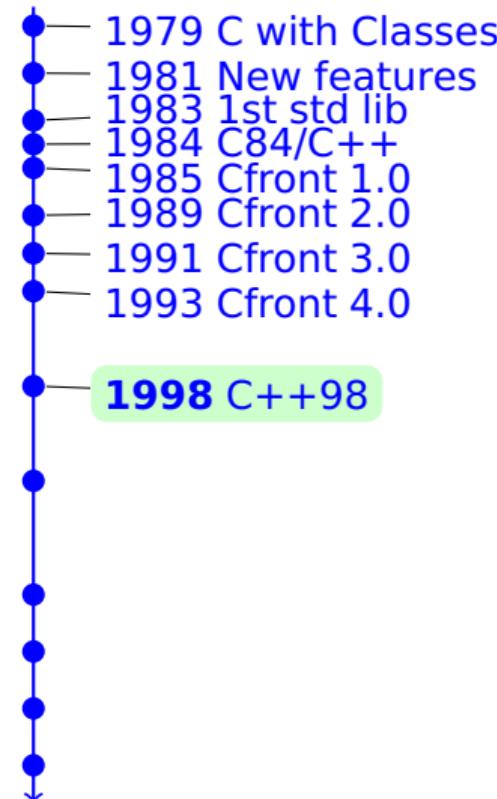
Language popularity  
oooooooooooo

Summary  
○○○○○

# 1998 - First ISO C++ standard

## Library additions:

- containers
- algorithms
- iterators
- bitset
- valarray
- auto\_ptr
- templated string
- iostream
- complex



C with Classes  
oooooooo

Cfront era  
oooooooooooooooooooo

Standardization time  
○○●○○○○

C++ future  
○○○○

(R)evolution!  
○○○○

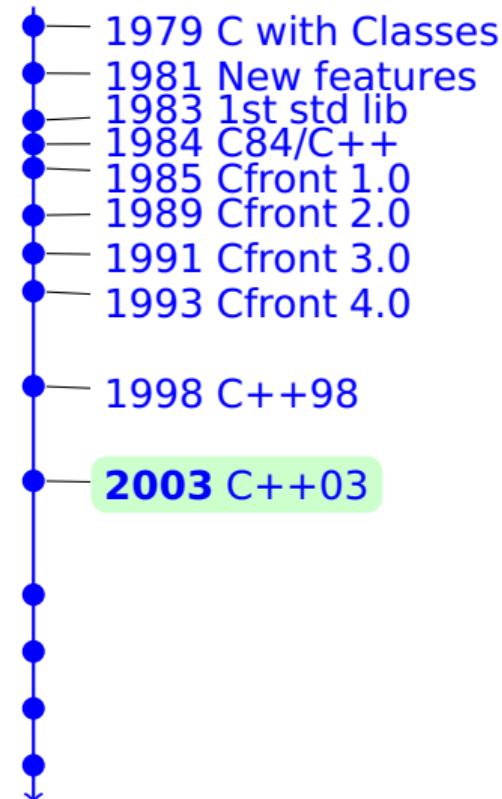
Language popularity  
oooooooooooo

Summary  
○○○○○

# C++03 - Bugfix release

Fixes:

- 125 defects fixed
- defect 69: incontiguous std::vector



C with Classes  
oooooooo

Cfront era  
oooooooooooooooooooo

Standardization time  
○○○●○○○○

C++ future  
○○○○

(R)evolution!  
○○○○○

Language popularity  
oooooooooooo

Summary  
○○○○○

# C++0x

*“C++11 feels like a new language”*

— Bjarne Stroustrup

C with Classes  
oooooooo

Cfront era  
oooooooooooooooooooo

Standardization time  
○○○●○○○○

C++ future  
○○○○

(R)evolution!  
○○○○○

Language popularity  
oooooooooooo

Summary  
○○○○○

# C++0x

*“C++11 feels like a new language”*

— Bjarne Stroustrup

*C++0x == C++11*

C with Classes  
oooooooo

Cfront era  
oooooooooooooooooooo

Standardization time  
○○○●○○○○

C++ future  
○○○○

(R)evolution!  
○○○○○

Language popularity  
oooooooooooo

Summary  
○○○○○

# C++0x

*“C++11 feels like a new language”*

— Bjarne Stroustrup

*C++0x == C++11 (for x = 0xB)*

C with Classes  
oooooooo

Cfront era  
oooooooooooooooooooo

Standardization time  
oooo●ooo

C++ future  
oooo

(R)evolution!  
ooooo

Language popularity  
oooooooooooo

Summary  
oooooo

# C++11

New language features:

C with Classes  
oooooooo

Cfront era  
oooooooooooooooooooo

Standardization time  
oooo●oooo

C++ future  
oooo

(R)evolution!  
ooooo

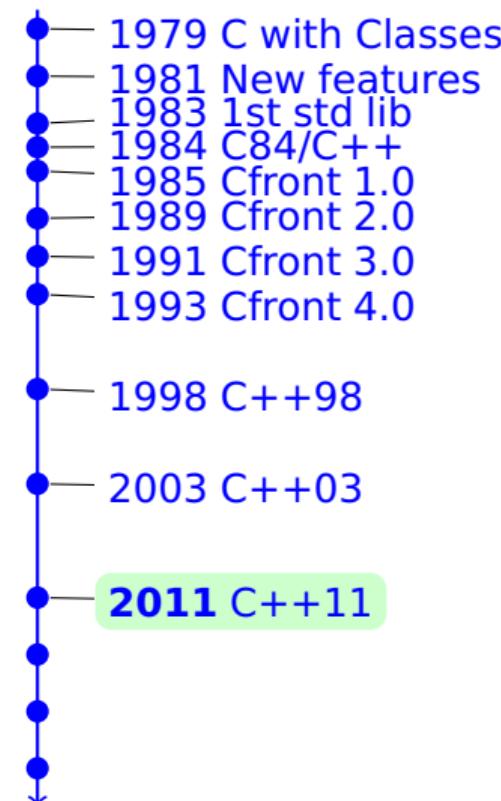
Language popularity  
oooooooooooo

Summary  
oooooo

# C++11

## New language features:

- auto and decltype
- default, delete, final, override keywords
- rvalue references
- move constructors / move assignment
- scoped enums
- constexpr
- list initialization, brace initializers
- delegating constructors
- nullptr
- type aliases
- smart pointers



C with Classes  
oooooooo

Cfront era  
oooooooooooooooooooo

Standardization time  
oooo●○○○

C++ future  
oooo

(R)evolution!  
ooooo

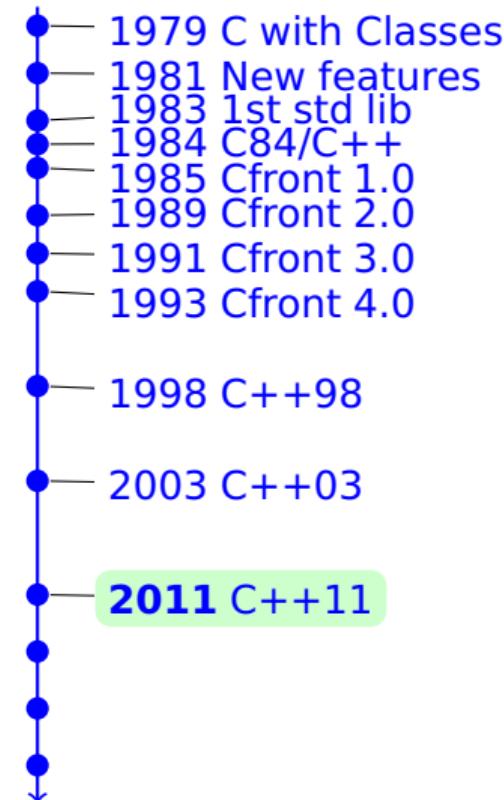
Language popularity  
oooooooooooo

Summary  
oooooo

# C++11

New language features:

- variadic templates
- user-defined literals
- attributes
- lambda expressions
- noexcept
- alignof and alignas
- multithreaded memory model
- thread-local storage
- range based for loop
- static assertions



C with Classes  
oooooooo

Cfront era  
oooooooooooooooooooo

Standardization time  
oooooooo●○○

C++ future  
oooo

(R)evolution!  
ooooo

Language popularity  
oooooooooooo

Summary  
oooooo

# C++14

New language features:

C with Classes  
oooooooo

Cfront era  
oooooooooooooooooooo

Standardization time  
oooooooo●○○

C++ future  
oooo

(R)evolution!  
ooooo

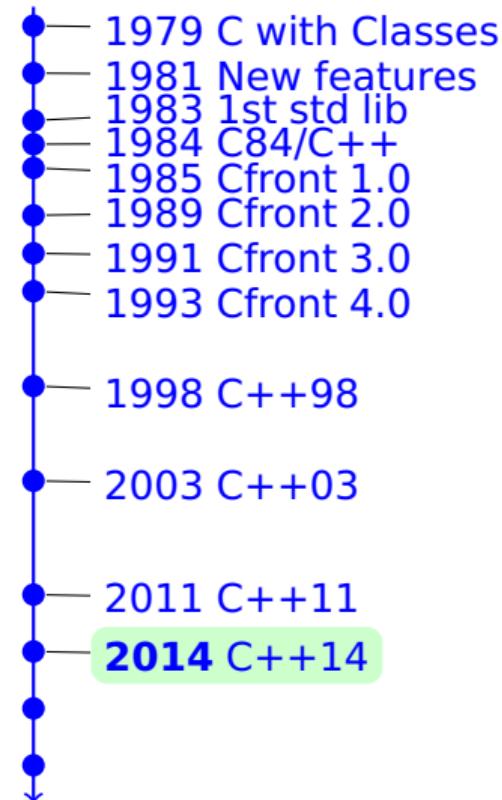
Language popularity  
oooooooooooo

Summary  
oooooo

# C++14

## New language features:

- generic lambdas
- lambda captures expressions
- function return type deduction
- alternate type deduction on declaration
- relaxed restrictions on constexpr functions
- variable templates
- binary literals
- digit separators
- deprecated attribute



C with Classes  
oooooooo

Cfront era  
oooooooooooooooooooo

Standardization time  
oooooooo●

C++ future  
oooo

(R)evolution!  
ooooo

Language popularity  
oooooooooooo

Summary  
oooooo

## Standardization - summary

- C++ standardization started in 1989

C with Classes  
oooooooo

Cfront era  
oooooooooooooooooooo

Standardization time  
oooooooo●

C++ future  
oooo

(R)evolution!  
ooooo

Language popularity  
oooooooooooo

Summary  
oooooo

## Standardization - summary

- C++ standardization started in 1989
- First ISO C++ standard: C++98

C with Classes  
oooooooo

Cfront era  
oooooooooooooooooooo

Standardization time  
oooooooo●

C++ future  
oooo

(R)evolution!  
ooooo

Language popularity  
oooooooooooo

Summary  
oooooo

## Standardization - summary

- C++ standardization started in 1989
- First ISO C++ standard: C++98
- Next standards: C++03, C++11, C++14

C with Classes  
oooooooo

Cfront era  
oooooooooooooooooooo

Standardization time  
oooooooo●

C++ future  
oooo

(R)evolution!  
ooooo

Language popularity  
oooooooooooo

Summary  
oooooo

## Standardization - summary

- C++ standardization started in 1989
- First ISO C++ standard: C++98
- Next standards: C++03, C++11, C++14
- Every bigger company had its own compiler

C with Classes  
oooooooo

Cfront era  
oooooooooooooooooooo

Standardization time  
oooooooo●

C++ future  
oooo

(R)evolution!  
ooooo

Language popularity  
oooooooooooo

Summary  
oooooo

## Standardization - summary

- C++ standardization started in 1989
- First ISO C++ standard: C++98
- Next standards: C++03, C++11, C++14
- Every bigger company had its own compiler
- Only 3 compilers fully support C++14:

C with Classes  
oooooooo

Cfront era  
oooooooooooooooooooo

Standardization time  
oooooooo●

C++ future  
oooo

(R)evolution!  
ooooo

Language popularity  
oooooooooooo

Summary  
oooooo

## Standardization - summary

- C++ standardization started in 1989
- First ISO C++ standard: C++98
- Next standards: C++03, C++11, C++14
- Every bigger company had its own compiler
- Only 3 compilers fully support C++14:
  - gcc

C with Classes  
oooooooo

Cfront era  
oooooooooooooooooooo

Standardization time  
oooooooo●

C++ future  
oooo

(R)evolution!  
ooooo

Language popularity  
oooooooooooo

Summary  
oooooo

## Standardization - summary

- C++ standardization started in 1989
- First ISO C++ standard: C++98
- Next standards: C++03, C++11, C++14
- Every bigger company had its own compiler
- Only 3 compilers fully support C++14:
  - gcc
  - clang

C with Classes  
oooooooo

Cfront era  
oooooooooooooooooooo

Standardization time  
oooooooo●

C++ future  
oooo

(R)evolution!  
ooooo

Language popularity  
oooooooooooo

Summary  
oooooo

## Standardization - summary

- C++ standardization started in 1989
- First ISO C++ standard: C++98
- Next standards: C++03, C++11, C++14
- Every bigger company had its own compiler
- Only 3 compilers fully support C++14:
  - gcc
  - clang
  - MSVC

C with Classes  
oooooooo

Cfront era  
oooooooooooooooooooo

Standardization time  
oooooooo●

C++ future  
oooo

(R)evolution!  
ooooo

Language popularity  
oooooooooooo

Summary  
oooooo

## Standardization - summary

- C++ standardization started in 1989
- First ISO C++ standard: C++98
- Next standards: C++03, C++11, C++14
- Every bigger company had its own compiler
- Only 3 compilers fully support C++14:
  - gcc
  - clang
  - MSVC
- full list: [http://en.cppreference.com/w/cpp/compiler\\_support](http://en.cppreference.com/w/cpp/compiler_support)

C with Classes  
oooooooo

Cfront era  
oooooooooooooooooooo

Standardization time  
oooooooo●

C++ future  
oooo

(R)evolution!  
ooooo

Language popularity  
oooooooooooo

Summary  
oooooo

## Standardization - summary

- C++ standardization started in 1989
- First ISO C++ standard: C++98
- Next standards: C++03, C++11, C++14
- Every bigger company had its own compiler
- Only 3 compilers fully support C++14:
  - gcc
  - clang
  - MSVC
- full list: [http://en.cppreference.com/w/cpp/compiler\\_support](http://en.cppreference.com/w/cpp/compiler_support)
- Number of users: about 3 000 000

C with Classes  
oooooooo

Cfront era  
ooooooooooooooo

Standardization time  
oooooooo

C++ future  
oooo

(R)evolution!  
oooo

Language popularity  
oooooooooooo

Summary  
oooooo

# C++ future



C with Classes  
oooooooo

Cfront era  
oooooooooooooooooooo

Standardization time  
oooooooooooo

C++ future  
●○○○

(R)evolution!  
○○○○

Language popularity  
oooooooooooo

Summary  
○○○○○

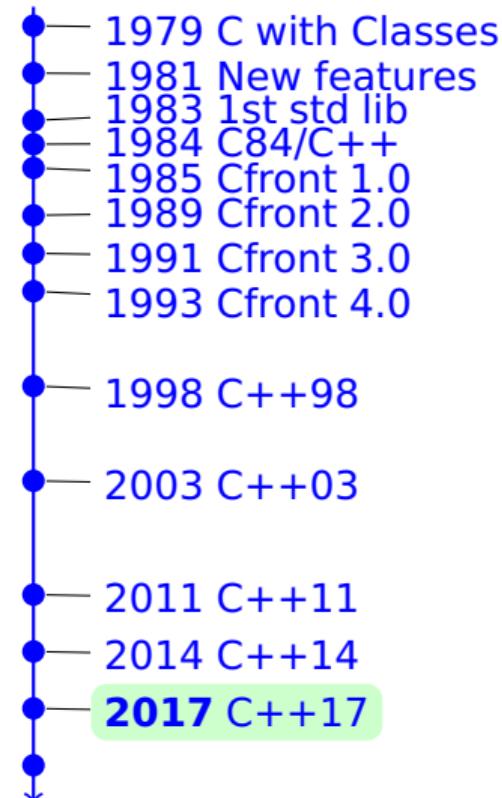
# C++17

New features:

# C++17

## New features:

- Filesystem TS
- Parallelism TS
- Library fundamentals TS
- `if constexpr(expression)`
- `auto` in templates
- structured bindings
- `if` and `switch` with initializer
- many, many more...



C with Classes  
oooooooo

Cfront era  
oooooooooooooooooooo

Standardization time  
oooooooooooo

C++ future  
○●○○

(R)evolution!  
ooooo

Language popularity  
oooooooooooo

Summary  
oooooo

# C++17

- Michael Wong - C++17, Will It Be Great Or Just OK
- Many planned features are out of scope
- Backward compatibility hamper the standardization of new features
- Stackoverflow list of C++17 features <sup>3</sup>

---

<sup>3</sup><http://stackoverflow.com/questions/38060436/what-are-the-new-features-in-c17>

C with Classes  
oooooooo

Cfront era  
oooooooooooooooooooo

Standardization time  
oooooooooooo

C++ future  
○○●○

(R)evolution!  
○○○○

Language popularity  
oooooooooooo

Summary  
○○○○○

# C++20



- 1979 C with Classes
- 1981 New features
- 1983 1st std lib
- 1984 C84/C++
- 1985 Cfront 1.0
- 1989 Cfront 2.0
- 1991 Cfront 3.0
- 1993 Cfront 4.0
- 1998 C++98
- 2003 C++03
- 2011 C++11
- 2014 C++14
- 2017 C++17
- **2020 C++20**

C with Classes  
oooooooo

Cfront era  
oooooooooooooooooooo

Standardization time  
oooooooooooo

C++ future  
○○○●

(R)evolution!  
○○○○

Language popularity  
oooooooooooo

Summary  
○○○○○

## C++ future - summary

- C++17 is almost ready

C with Classes  
oooooooo

Cfront era  
oooooooooooooooooooo

Standardization time  
oooooooooooo

C++ future  
○○○●

(R)evolution!  
○○○○

Language popularity  
oooooooooooo

Summary  
○○○○○

## C++ future - summary

- C++17 is almost ready
- Future versions are planned to be released every 3 years

C with Classes  
oooooooo

Cfront era  
oooooooooooooooooooo

Standardization time  
oooooooooooo

C++ future  
○○○●

(R)evolution!  
○○○○

Language popularity  
oooooooooooo

Summary  
○○○○○

## C++ future - summary

- C++17 is almost ready
- Future versions are planned to be released every 3 years
- Next planned version: C++20 (minor)

C with Classes  
oooooooo

Cfront era  
oooooooooooooooooooo

Standardization time  
oooooooooooo

C++ future  
○○○●

(R)evolution!  
○○○○

Language popularity  
oooooooooooo

Summary  
○○○○○

## C++ future - summary

- C++17 is almost ready
- Future versions are planned to be released every 3 years
- Next planned version: C++20 (minor)
- Backward compatibility hamper the standardization of new features

C with Classes  
oooooooo

Cfront era  
oooooooooooooooooooo

Standardization time  
oooooooooooo

C++ future  
○○○●

(R)evolution!  
○○○○

Language popularity  
oooooooooooo

Summary  
○○○○○

## C++ future - summary

- C++17 is almost ready
- Future versions are planned to be released every 3 years
- Next planned version: C++20 (minor)
- Backward compatibility hamper the standardization of new features
- gcc and clang already support many of C++17 features

C with Classes  
oooooooo

Cfront era  
oooooooooooooooooooo

Standardization time  
oooooooooooo

C++ future  
○○○●

(R)evolution!  
○○○○

Language popularity  
oooooooooooo

Summary  
○○○○○

## C++ future - summary

- C++17 is almost ready
- Future versions are planned to be released every 3 years
- Next planned version: C++20 (minor)
- Backward compatibility hamper the standardization of new features
- gcc and clang already support many of C++17 features
- MSVC is behind them

C with Classes  
oooooooo

Cfront era  
oooooooooooooooooooo

Standardization time  
oooooooooooo

C++ future  
oooo

(R)evolution!  
●oooo

Language popularity  
oooooooooooo

Summary  
oooooo

# (R)evolution!

Evolution of C++ vs Revolution of C++

C with Classes  
oooooooo

Cfront era  
oooooooooooooooooooo

Standardization time  
oooooooooooo

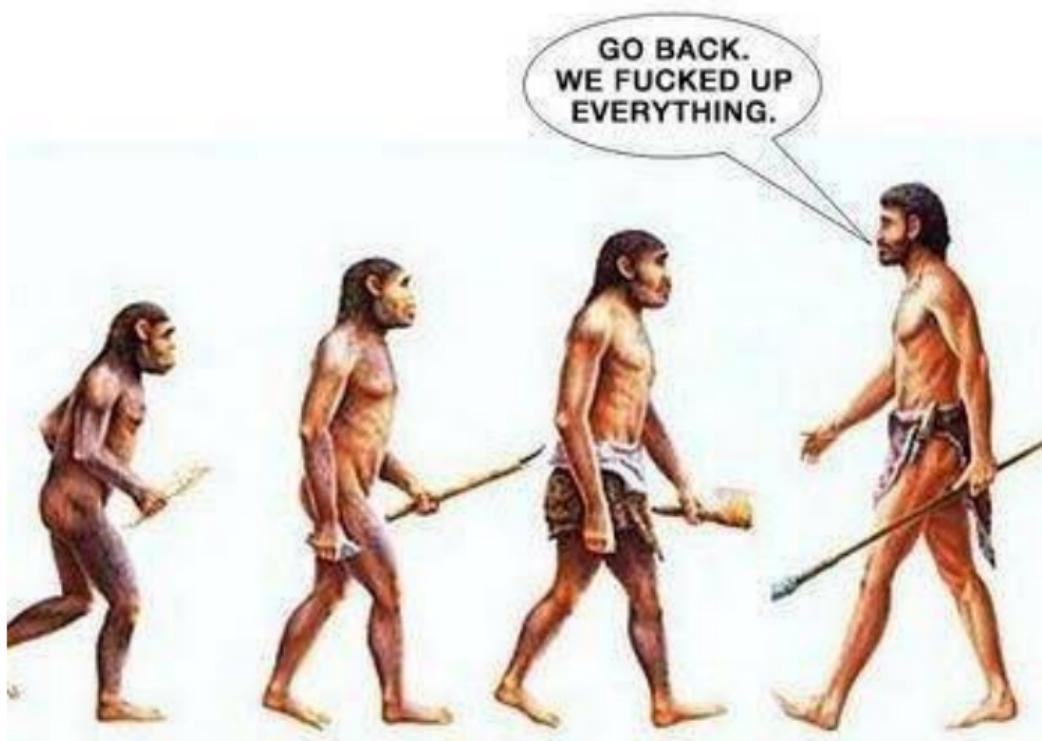
C++ future  
oooo

(R)evolution!  
●oooo

Language popularity  
oooooooooooo

Summary  
oooooo

# (R)evolution!



C with Classes  
oooooooo

Cfront era  
oooooooooooooooooooo

Standardization time  
oooooooo

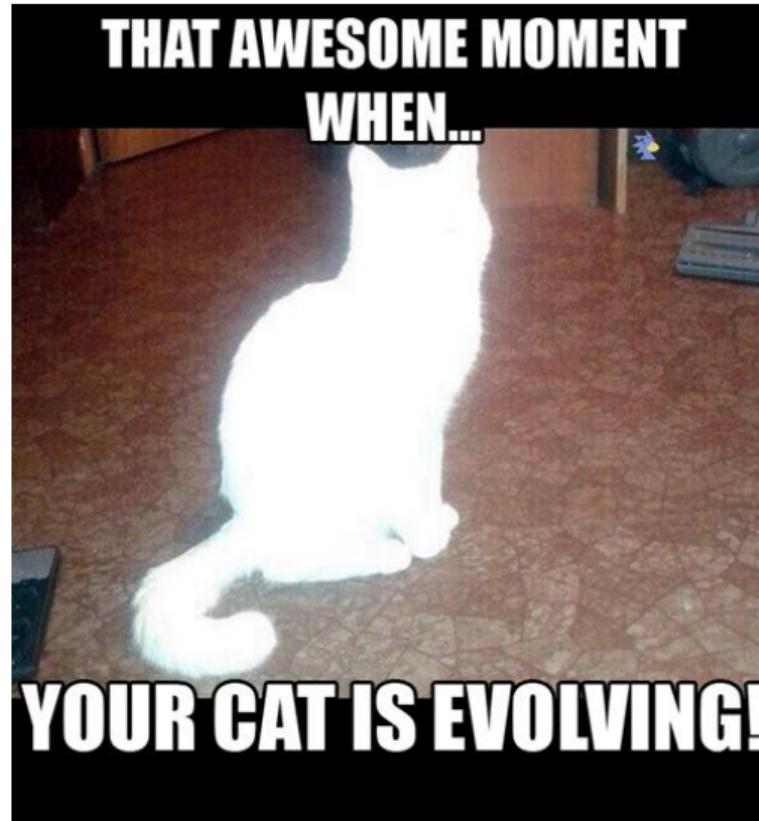
C++ future  
oooo

(R)evolution!  
●oooo

Language popularity  
oooooooooooo

Summary  
oooooo

# (R)evolution!



C with Classes  
oooooooo

Cfront era  
oooooooooooooooooooo

Standardization time  
oooooooooooo

C++ future  
oooo

(R)evolution!  
o●ooo

Language popularity  
oooooooooooo

Summary  
oooooo

# C++ is becoming more and more complicated...



C with Classes  
oooooooo

Cfront era  
oooooooooooooooooooo

Standardization time  
oooooooooooo

C++ future  
oooo

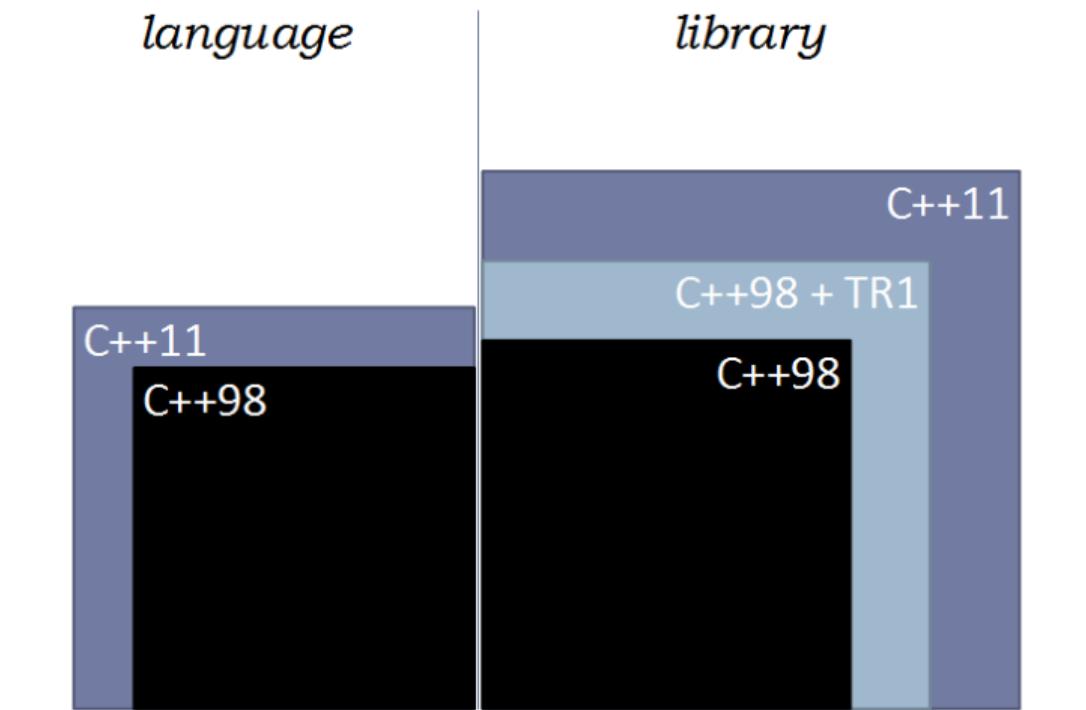
(R)evolution!  
○○●○○

Language popularity  
oooooooooooo

Summary  
ooooooo

# stdlib is poor

- Some people say that C++ standard library is small...



C with Classes  
oooooooo

Cfront era  
oooooooooooooooooooo

Standardization time  
oooooooooooo

C++ future  
oooo

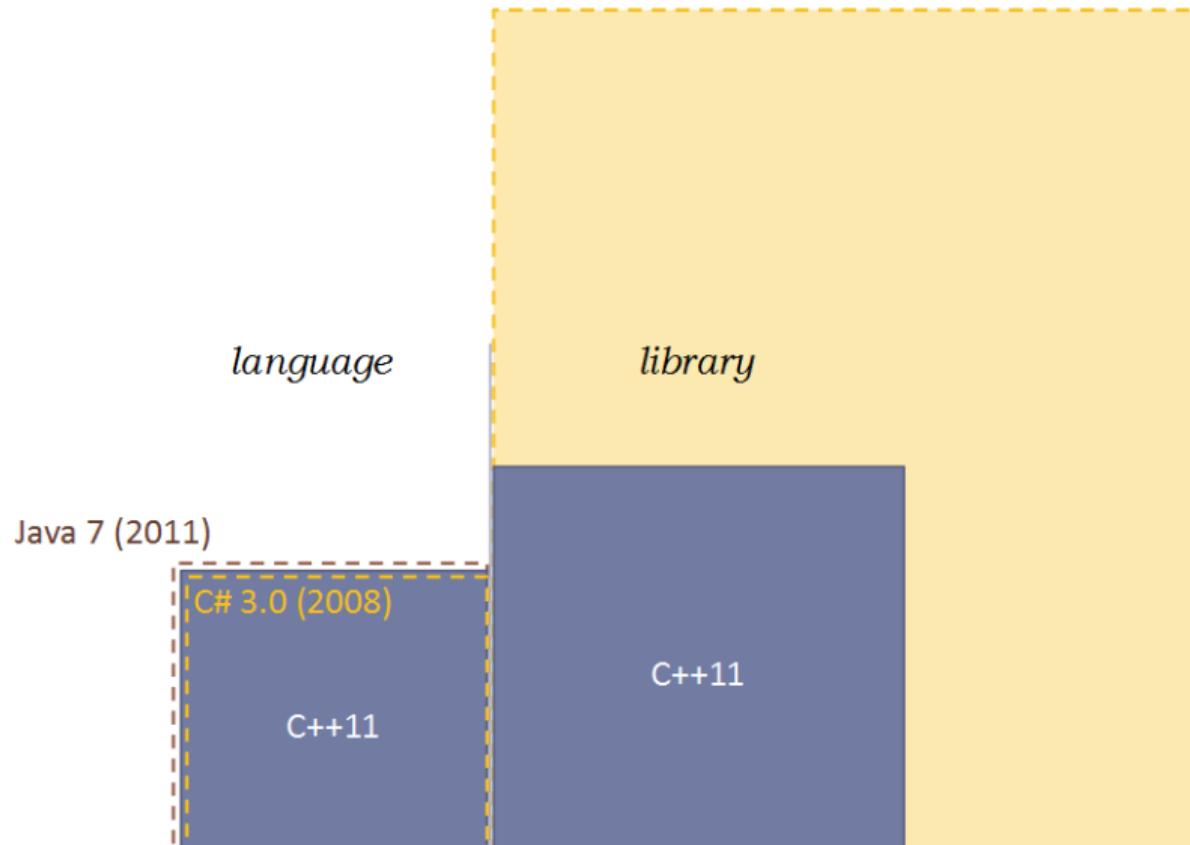
(R)evolution!  
○○●○○

Language popularity  
oooooooooooo

Summary  
ooooooo

# stdlib is poor

- Some people say that C++ standard library is small...
- In comparison to other languages



C with Classes  
oooooooo

Cfront era  
oooooooooooooooooooo

Standardization time  
oooooooooooo

C++ future  
oooo

(R)evolution!  
○○●○○

Language popularity  
oooooooooooo

Summary  
ooooooo

# stdlib is poor

- Some people say that C++ standard library is small...
- In comparison to other languages
- Examples from presentation "One C++" by Herb Sutter

*language*

*library*

Java SE 7

2008 .NET FX (only)



C with Classes  
oooooooo

Cfront era  
oooooooooooooooooooo

Standardization time  
oooooooo

C++ future  
oooo

(R)evolution!  
○○●○○

Language popularity  
oooooooooooo

Summary  
oooooo

# stdlib is poor

- Some people say that C++ standard library is small...
- In comparison to other languages
- Examples from presentation "One C++" by Herb Sutter
- But it will grow in next standards

Han ma boo-kee, keelee ka-lya dooka. Wadja da boolya na 1.9 Megabytes



Bloated JabbaScript  
Frameworks

*The Smuggler's Guide*

C with Classes  
oooooooo

Cfront era  
oooooooooooooooooooo

Standardization time  
oooooooooooo

C++ future  
oooo

(R)evolution!  
○○○●○

Language popularity  
oooooooooooo

Summary  
ooooooo

# C++ standard implementation delays

<b>Version</b>	<b>Standard</b>	<b>First implementation</b>
C84	"the ARM" - 1989	-
C++98	IX 1998	2003 (EDG + Dinkumware)
C++03	X 2003	?
C++11	IX 2011	IV 2013 (clang3.3)
C++14	III 2014	XI 2013 (clang 3.4)
C++17	?	?

C with Classes  
oooooooo

Cfront era  
oooooooooooooooooooo

Standardization time  
oooooooooooo

C++ future  
oooo

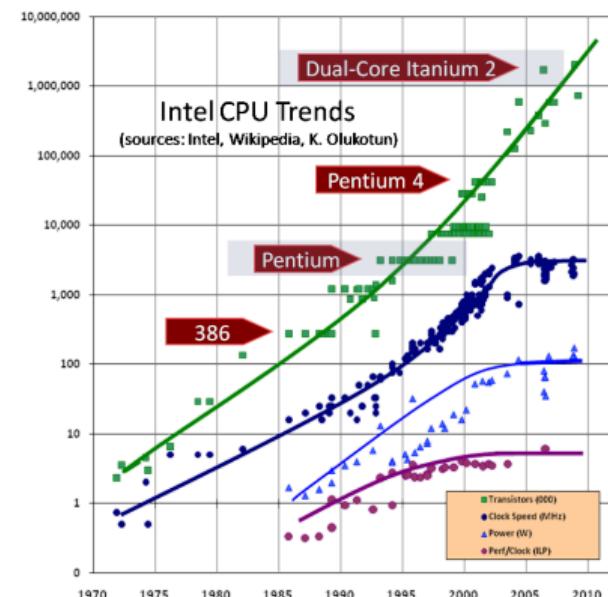
(R)evolution!  
oooo●

Language popularity  
oooooooooooo

Summary  
ooooooo

# Free lunch is over

- Moore's law is no applicable anymore



[http://www.gotw.ca/publications/  
concurrency-ddj.htm](http://www.gotw.ca/publications/concurrency-ddj.htm)

C with Classes  
oooooooo

Cfront era  
oooooooooooooooooooo

Standardization time  
oooooooooooo

C++ future  
oooo

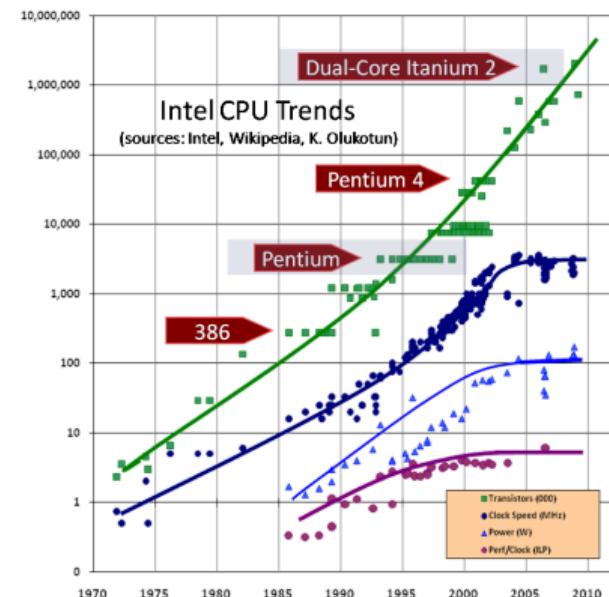
(R)evolution!  
oooo●

Language popularity  
oooooooooooo

Summary  
ooooooo

# Free lunch is over

- Moore's law is no applicable anymore
- Processor speed doesn't rise



[http://www.gotw.ca/publications/  
concurrency-ddj.htm](http://www.gotw.ca/publications/concurrency-ddj.htm)

C with Classes  
oooooooo

Cfront era  
oooooooooooooooooooo

Standardization time  
oooooooo

C++ future  
oooo

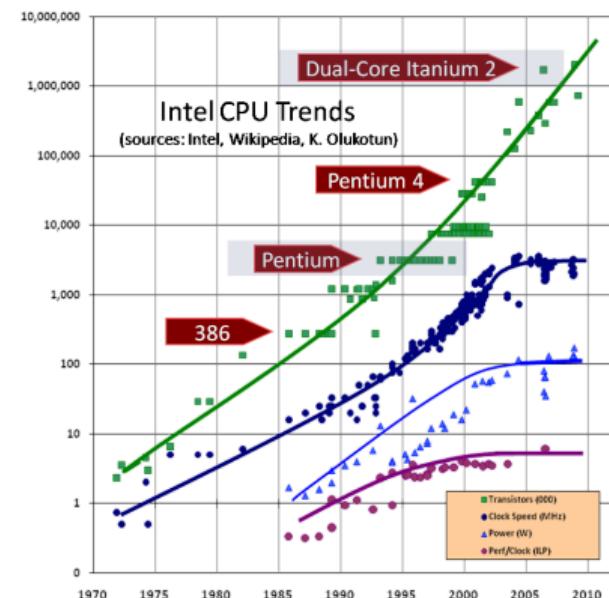
(R)evolution!  
oooo●

Language popularity  
oooooooooooo

Summary  
oooooo

# Free lunch is over

- Moore's law is no applicable anymore
- Processor speed doesn't rise
- We must go into concurrency



[http://www.gotw.ca/publications/  
concurrency-ddj.htm](http://www.gotw.ca/publications/concurrency-ddj.htm)

C with Classes  
oooooooo

Cfront era  
oooooooooooooooooooo

Standardization time  
oooooooo

C++ future  
oooo

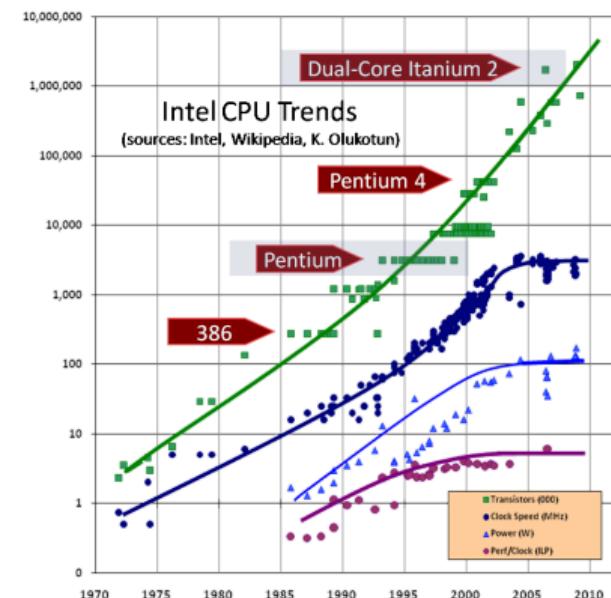
(R)evolution!  
oooo●

Language popularity  
oooooooooooo

Summary  
oooooo

# Free lunch is over

- Moore's law is no applicable anymore
- Processor speed doesn't rise
- We must go into concurrency
- We must write multithreaded apps



[http://www.gotw.ca/publications/  
concurrency-ddj.htm](http://www.gotw.ca/publications/concurrency-ddj.htm)

C with Classes  
oooooooo

Cfront era  
oooooooooooooooooooo

Standardization time  
oooooooo

C++ future  
oooo

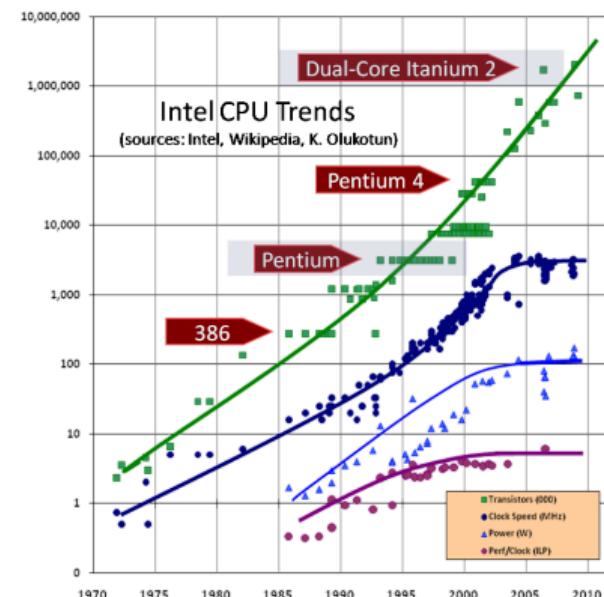
(R)evolution!  
oooo●

Language popularity  
oooooooooooo

Summary  
oooooo

# Free lunch is over

- Moore's law is no applicable anymore
- Processor speed doesn't rise
- We must go into concurrency
- We must write multithreaded apps
- We must write efficient and effective code



[http://www.gotw.ca/publications/  
concurrency-ddj.htm](http://www.gotw.ca/publications/concurrency-ddj.htm)

C with Classes  
oooooooo

Cfront era  
oooooooooooooooooooo

Standardization time  
oooooooooooo

C++ future  
oooo

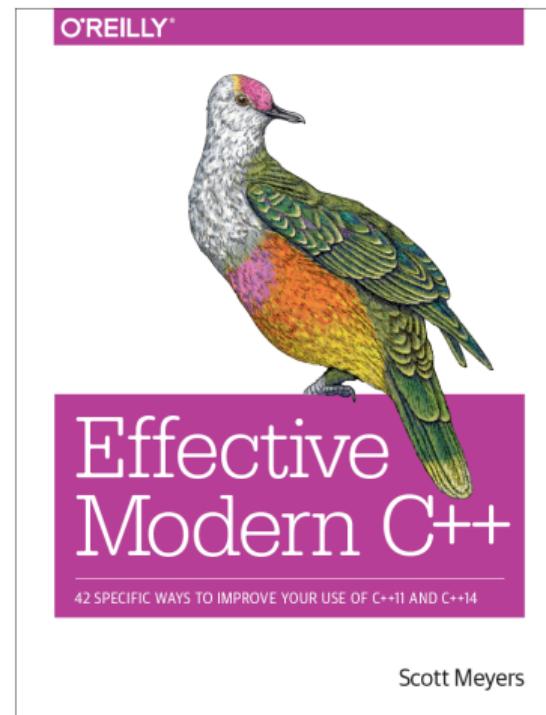
(R)evolution!  
oooo●

Language popularity  
oooooooooooo

Summary  
ooooooo

# Free lunch is over

- Moore's law is no applicable anymore
- Processor speed doesn't rise
- We must go into concurrency
- We must write multithreaded apps
- We must write efficient and effective code
- Modern C++ facilitate above needs



(This book cover is real)

C with Classes  
oooooooo

Cfront era  
oooooooooooooooooooo

Standardization time  
oooooooooooo

C++ future  
oooo

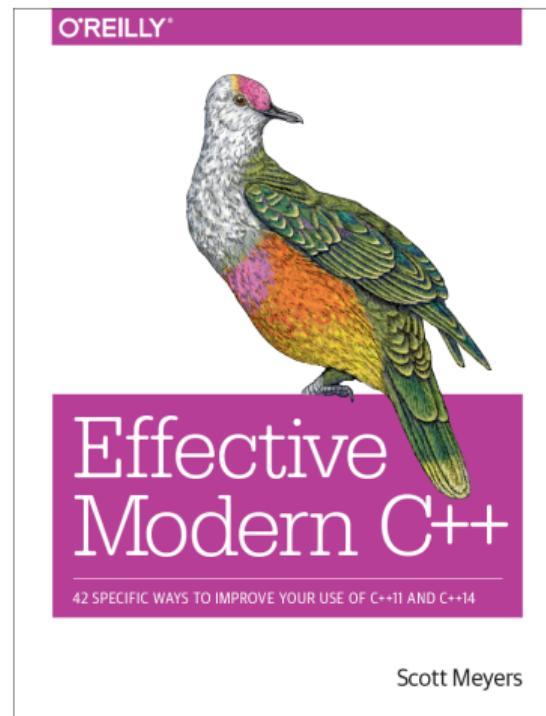
(R)evolution!  
oooo●

Language popularity  
oooooooooooo

Summary  
ooooooo

# Free lunch is over

- Moore's law is no applicable anymore
- Processor speed doesn't rise
- We must go into concurrency
- We must write multithreaded apps
- We must write efficient and effective code
- Modern C++ facilitate above needs
- VM languages will not be faster than C++



(This book cover is real)

C with Classes  
oooooooo

Cfront era  
oooooooooooooooooooo

Standardization time  
oooooooooooo

C++ future  
oooo

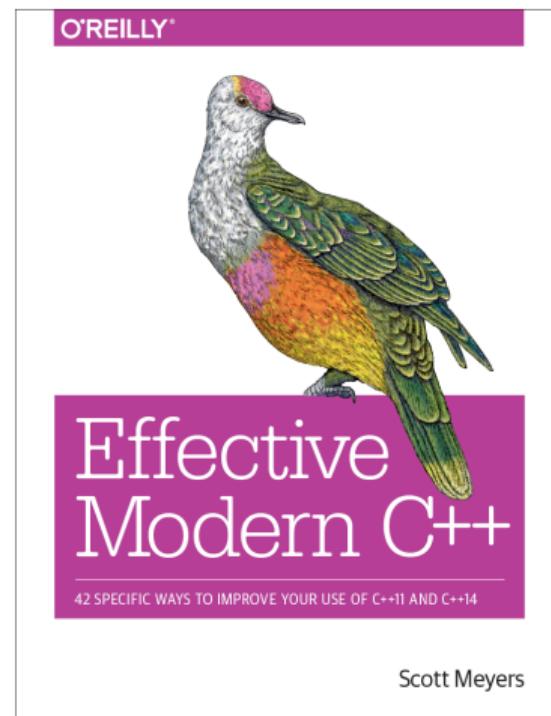
(R)evolution!  
oooo●

Language popularity  
oooooooooooo

Summary  
ooooooo

# Free lunch is over

- Moore's law is no applicable anymore
- Processor speed doesn't rise
- We must go into concurrency
- We must write multithreaded apps
- We must write efficient and effective code
- Modern C++ facilitate above needs
- VM languages will not be faster than C++
- More and more mobile apps are written in C++



(This book cover is real)

# Free lunch is over

- Moore's law is no applicable anymore
- Processor speed doesn't rise
- We must go into concurrency
- We must write multithreaded apps
- We must write efficient and effective code
- Modern C++ facilitate above needs
- VM languages will not be faster than C++
- More and more mobile apps are written in C++
- Can C perform better?



Bartosz 'BaSz' Szurgot  
C++ vs. C: The embedded perspective  
code::dive 2015

C with Classes  
oooooooo

Cfront era  
oooooooooooooooooooo

Standardization time  
oooooooooooo

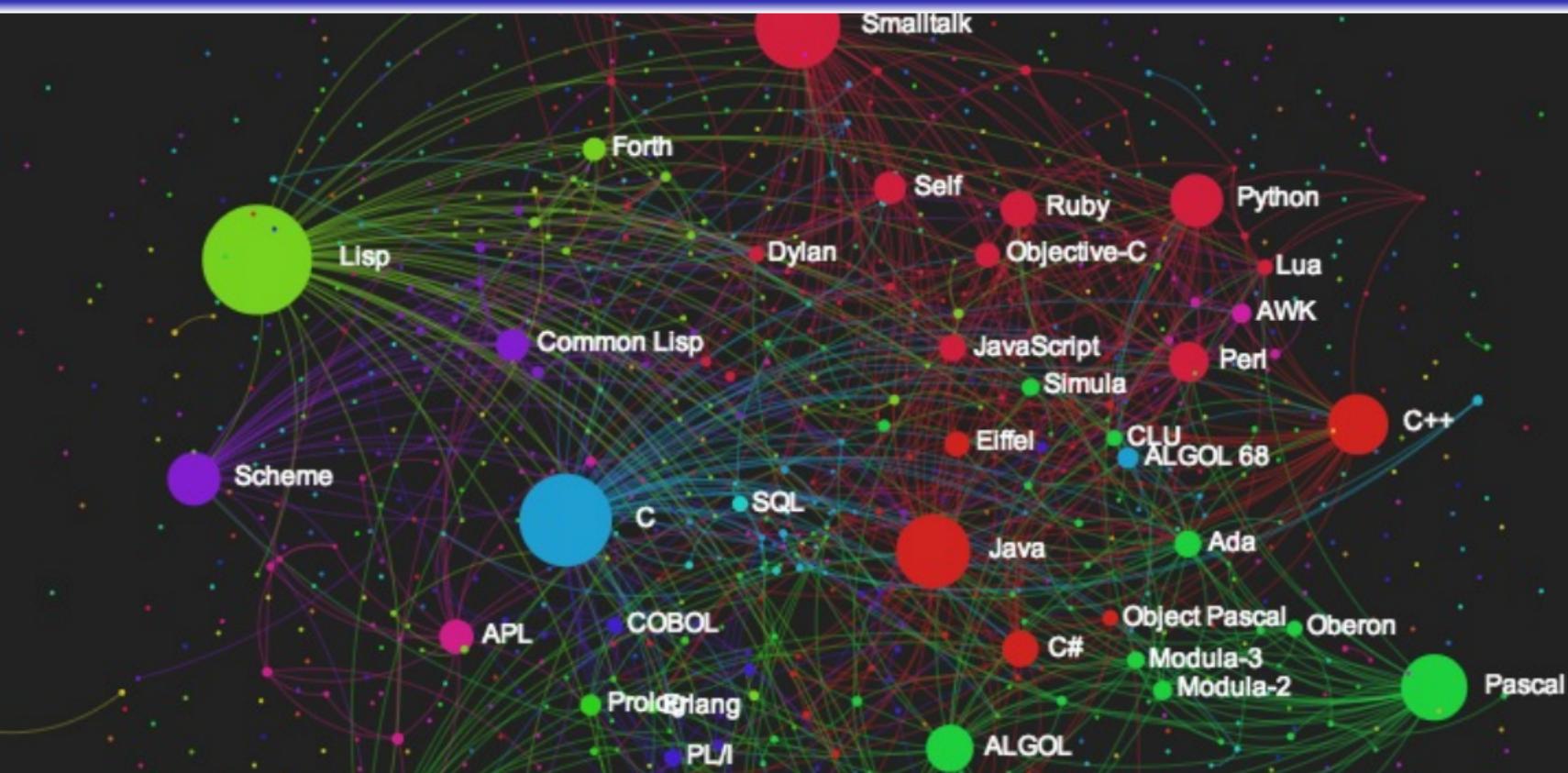
C++ future  
oooo

(R)evolution!  
oooo

Language popularity  
oooooooooooo

Summary  
oooooo

# Language popularity



C with Classes  
oooooooo

Cfront era  
oooooooooooooooooooo

Standardization time  
oooooooooooo

C++ future  
oooo

(R)evolution!  
ooooo

Language popularity  
●oooooooooooo

Summary  
oooooo

# What is the most popular programming language?

C with Classes  
oooooooo

Cfront era  
oooooooooooooooooooo

Standardization time  
oooooooooooo

C++ future  
oooo

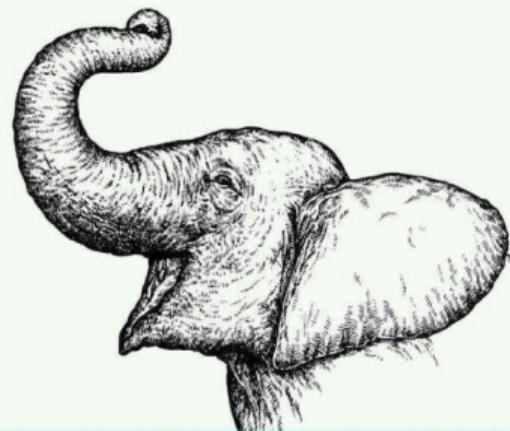
(R)evolution!  
ooooo

Language popularity  
●oooooooooooo

Summary  
oooooo

# What is the most popular programming language?

*The answer to every programming question ever conceived*



# It Depends

*The Definitive Guide*

O RLY?

@ThePracticalDev

C with Classes  
oooooooo

Cfront era  
oooooooooooooooooooo

Standardization time  
oooooooooooo

C++ future  
oooo

(R)evolution!  
ooooo

Language popularity  
○●oooooooooooo

Summary  
oooooo

# C++ users

Date	Estimated users
1979	1
1980	16
1981	38
1982	85
1983	??+2
1984	??+50
1985	500
1986	2 000
1987	4 000
1988	15 000
1989	50 000
1990	150 000
1991	400 000

C with Classes  
oooooooo

Cfront era  
oooooooooooooooooooo

Standardization time  
oooooooooooo

C++ future  
oooo

(R)evolution!  
ooooo

Language popularity  
○●oooooooooooo

Summary  
oooooo

# C++ users

Date	Estimated users
1979	1
1980	16
1981	38
1982	85
1983	??+2
1984	??+50
1985	500
1986	2 000
1987	4 000
1988	15 000
1989	50 000
1990	150 000
1991	400 000

Main users:

- Bjarne,

C with Classes  
oooooooo

Cfront era  
oooooooooooooooooooo

Standardization time  
oooooooooooo

C++ future  
oooo

(R)evolution!  
ooooo

Language popularity  
○●oooooooooooo

Summary  
oooooo

# C++ users

Date	Estimated users
1979	1
1980	16
1981	38
1982	85
1983	??+2
1984	??+50
1985	500
1986	2 000
1987	4 000
1988	15 000
1989	50 000
1990	150 000
1991	400 000

Main users:

- Bjarne,
- Bjarne's colleagues from AT&T Bell Labs,

C with Classes  
oooooooo

Cfront era  
oooooooooooooooooooo

Standardization time  
oooooooooooo

C++ future  
oooo

(R)evolution!  
ooooo

Language popularity  
○●oooooooooooo

Summary  
oooooo

# C++ users

Date	Estimated users
1979	1
1980	16
1981	38
1982	85
1983	??+2
1984	??+50
1985	500
1986	2 000
1987	4 000
1988	15 000
1989	50 000
1990	150 000
1991	400 000

Main users:

- Bjarne,
- Bjarne's colleagues from AT&T Bell Labs,
- universities,

C with Classes  
oooooooo

Cfront era  
oooooooooooooooooooo

Standardization time  
oooooooooooo

C++ future  
oooo

(R)evolution!  
ooooo

Language popularity  
○●oooooooooooo

Summary  
oooooo

# C++ users

Date	Estimated users
1979	1
1980	16
1981	38
1982	85
1983	??+2
1984	??+50
1985	500
1986	2 000
1987	4 000
1988	15 000
1989	50 000
1990	150 000
1991	400 000

Main users:

- Bjarne,
- Bjarne's colleagues from AT&T Bell Labs,
- universities,
- HP, IBM, AT&T, DEC,

C with Classes  
oooooooo

Cfront era  
oooooooooooooooooooo

Standardization time  
oooooooooooo

C++ future  
oooo

(R)evolution!  
ooooo

Language popularity  
○●oooooooooooo

Summary  
oooooo

# C++ users

Date	Estimated users
1979	1
1980	16
1981	38
1982	85
1983	??+2
1984	??+50
1985	500
1986	2 000
1987	4 000
1988	15 000
1989	50 000
1990	150 000
1991	400 000

Main users:

- Bjarne,
- Bjarne's colleagues from AT&T Bell Labs,
- universities,
- HP, IBM, AT&T, DEC,
- Borland,

C with Classes  
oooooooo

Cfront era  
oooooooooooooooooooo

Standardization time  
oooooooooooo

C++ future  
oooo

(R)evolution!  
ooooo

Language popularity  
○●oooooooooooo

Summary  
oooooo

# C++ users

Date	Estimated users
1979	1
1980	16
1981	38
1982	85
1983	??+2
1984	??+50
1985	500
1986	2 000
1987	4 000
1988	15 000
1989	50 000
1990	150 000
1991	400 000

Main users:

- Bjarne,
- Bjarne's colleagues from AT&T Bell Labs,
- universities,
- HP, IBM, AT&T, DEC,
- Borland,
- Later: Microsoft, Apple,

C with Classes  
oooooooo

Cfront era  
oooooooooooooooooooo

Standardization time  
oooooooooooo

C++ future  
oooo

(R)evolution!  
ooooo

Language popularity  
○●oooooooooooo

Summary  
oooooo

# C++ users

Date	Estimated users
1979	1
1980	16
1981	38
1982	85
1983	??+2
1984	??+50
1985	500
1986	2 000
1987	4 000
1988	15 000
1989	50 000
1990	150 000
1991	400 000

Main users:

- Bjarne,
- Bjarne's colleagues from AT&T Bell Labs,
- universities,
- HP, IBM, AT&T, DEC,
- Borland,
- Later: Microsoft, Apple,
- Now: Google, Facebook

C with Classes  
oooooooo

Cfront era  
oooooooooooooooooooo

Standardization time  
oooooooo

C++ future  
oooo

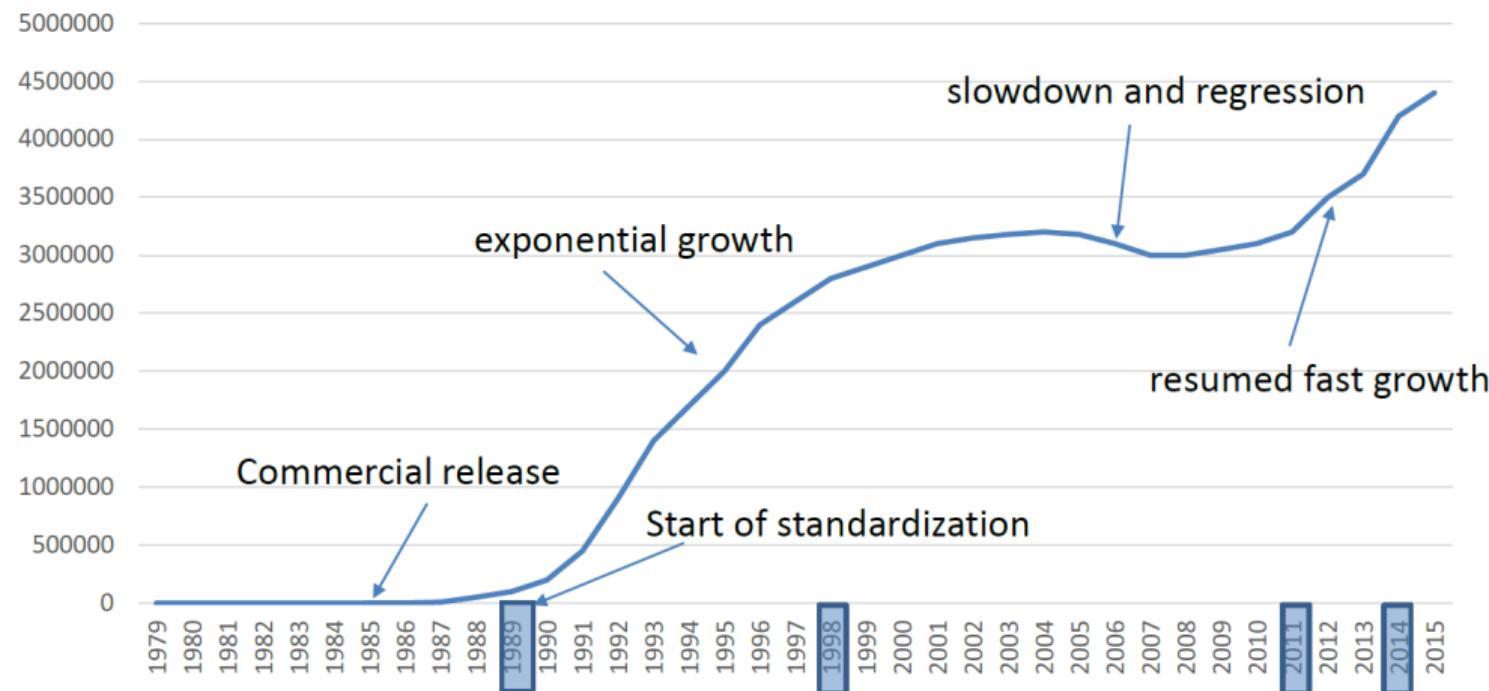
(R)evolution!  
oooo

Language popularity  
oo●oooooooo

Summary  
oooooo

# C++ users

#C++ users (approximate, with interpolation)



<sup>3</sup>Bjarne Stroustrup - *The Evolution of C++* - [https://www.youtube.com/watch?v=\\_wzc7a3Mc0s](https://www.youtube.com/watch?v=_wzc7a3Mc0s)

C with Classes  
oooooooo

Cfront era  
oooooooooooooooooooo

Standardization time  
oooooooooooo

C++ future  
oooo

(R)evolution!  
ooooo

Language popularity  
oooo●oooooooo

Summary  
oooooo

# TIOBE - market share

## TIOBE index

The **TIOBE Programming Community** index is an indicator of the popularity of programming languages. The index is updated once a month. Popular search engines such as **Google, Bing, Yahoo!, Wikipedia, Amazon, YouTube** and **Baidu** are used to calculate the ratings. Search phrase is "**language programming**"

Webpage: [http://www.tiobe.com/tiobe-index//](http://www.tiobe.com/tiobe-index/)

C with Classes  
oooooooo

Cfront era  
oooooooooooooooooooo

Standardization time  
oooooooooooo

C++ future  
oooo

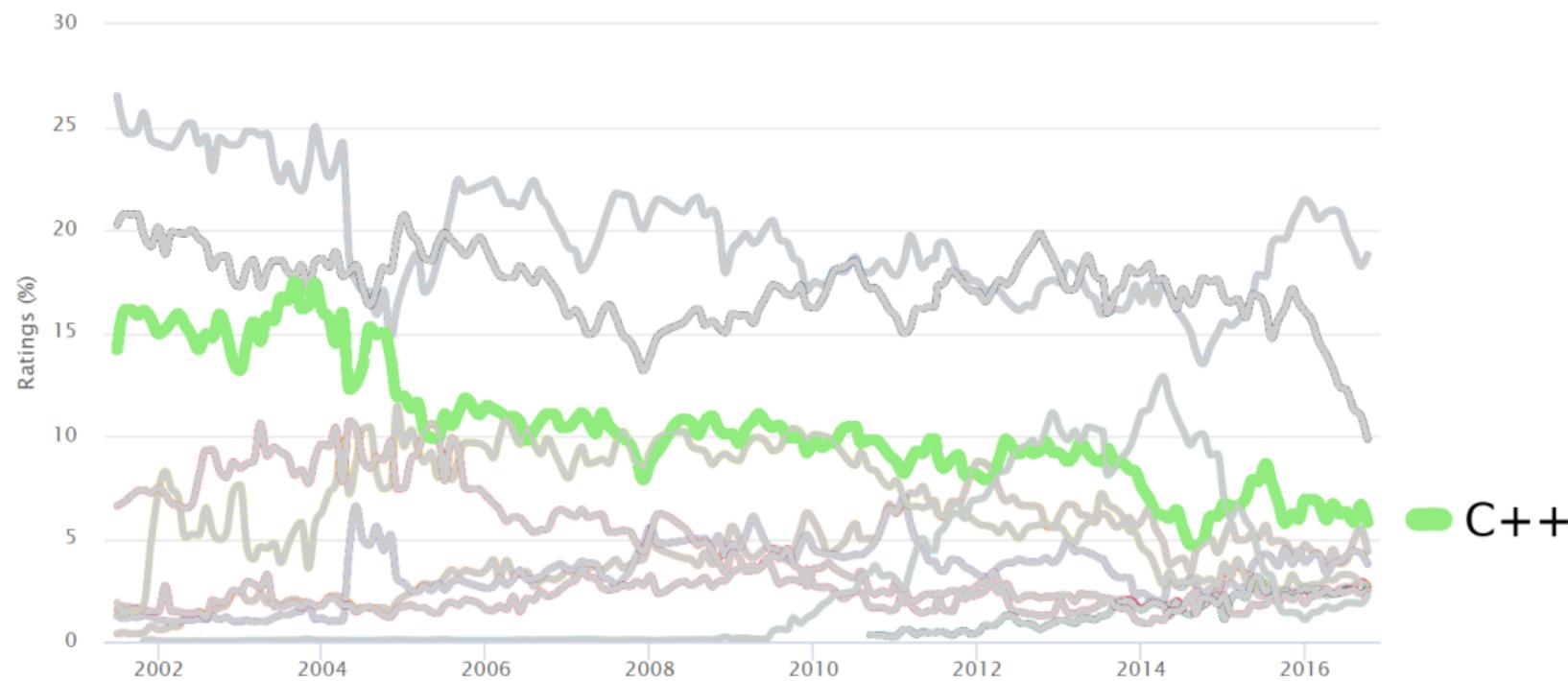
(R)evolution!  
oooo

Language popularity  
oooo•oooooooo

Summary  
oooooo

# TIOBE - market share

Source: [www.tiobe.com](http://www.tiobe.com)



C with Classes  
oooooooo

Cfront era  
oooooooooooooooooooo

Standardization time  
oooooooo

C++ future  
oooo

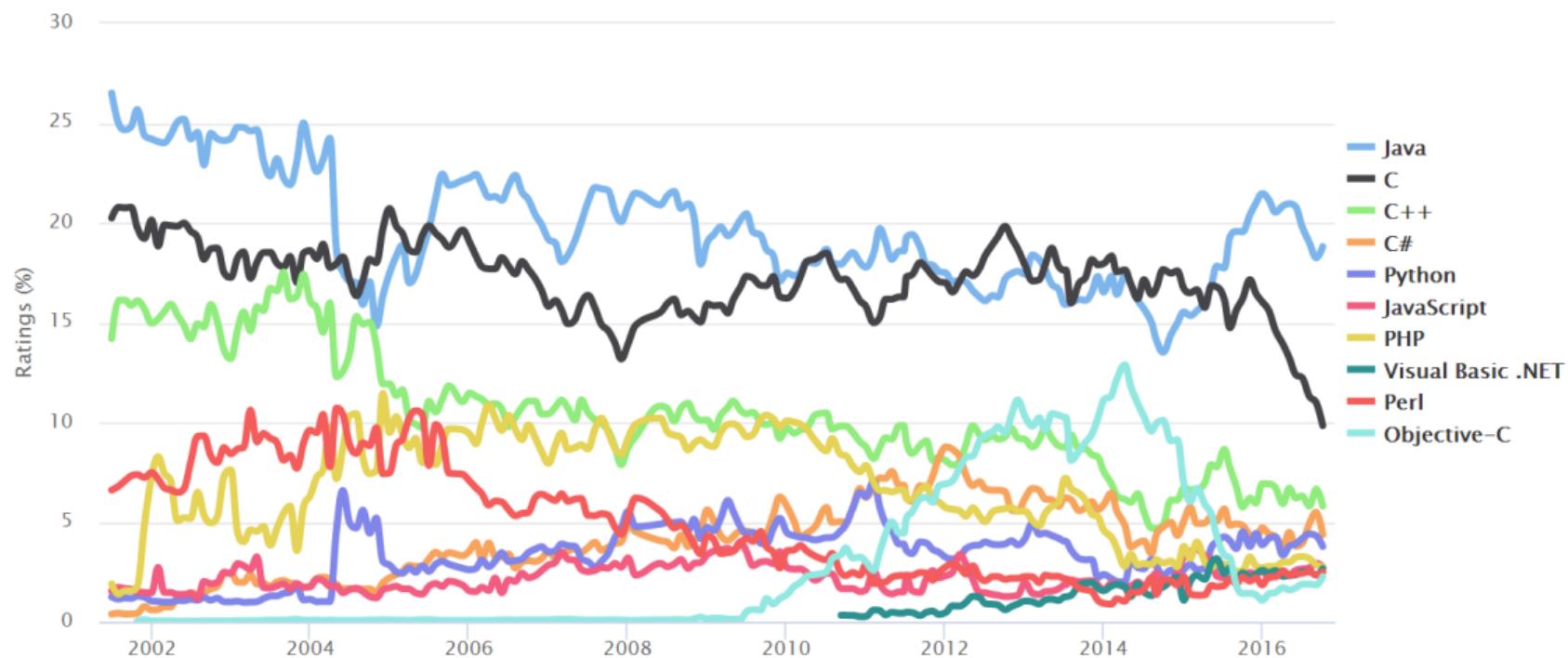
(R)evolution!  
oooo

Language popularity  
ooo•oooooooo

Summary  
oooooo

# TIOBE - market share

Source: [www.tiobe.com](http://www.tiobe.com)



C with Classes  
ooooooooCfront era  
ooooooooooooooooooooStandardization time  
ooooooooooooC++ future  
oooo(R)evolution!  
ooooooLanguage popularity  
oooo●ooooooooSummary  
ooooooo

# TIOBE - market share

Oct 2016	Oct 2015	Change	Programming Language	Ratings	Change
1	1		Java	18.799%	-0.74%
2	2		C	9.835%	-6.35%
3	3		C++	5.797%	+0.05%
4	4		C#	4.367%	-0.46%
5	5		Python	3.775%	-0.74%
6	8	▲	JavaScript	2.751%	+0.46%
7	6	▼	PHP	2.741%	+0.18%
8	7	▼	Visual Basic .NET	2.660%	+0.20%
9	9		Perl	2.495%	+0.25%
10	14	▲	Objective-C	2.263%	+0.84%

C with Classes  
oooooooo

Cfront era  
oooooooooooooooooooo

Standardization time  
oooooooooooo

C++ future  
oooo

(R)evolution!  
ooooo

Language popularity  
oooo●oooooooo

Summary  
oooooo

# PYPL

## PYPL index

**The PYPL PopularitY of Programming Language Index** is created by analyzing how often **language tutorials** are searched on Google: the more a language tutorial is searched, the more popular the language is assumed to be. It is a leading indicator. The raw data comes from **Google Trends**.

Webpage: <http://pypl.github.io/PYPL.html>

C with Classes  
oooooooo

Cfront era  
oooooooooooooooooooo

Standardization time  
oooooooooooo

C++ future  
oooo

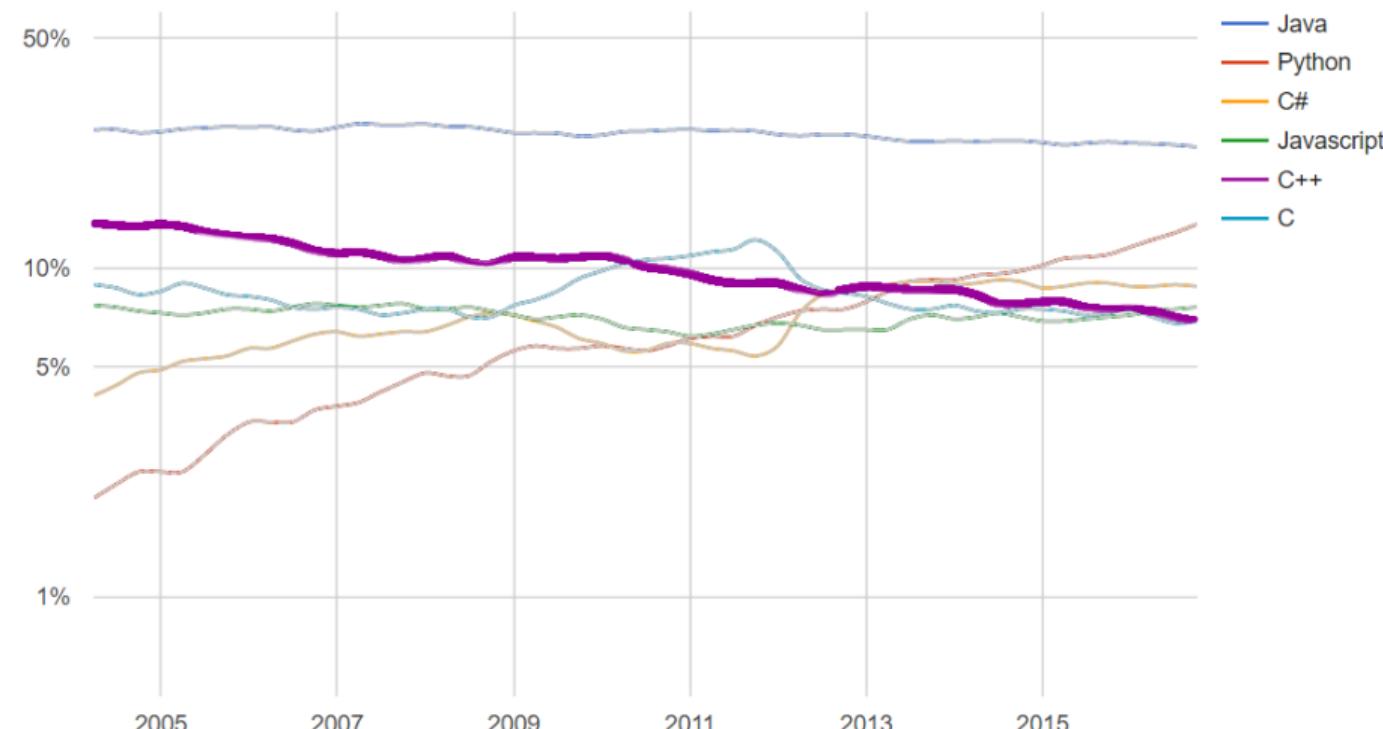
(R)evolution!  
oooo

Language popularity  
oooo●oooooooo

Summary  
ooooooo

# PYPL

## PYPL PopularitY of Programming Language



C with Classes  
oooooooo

Cfront era  
oooooooooooooooooooo

Standardization time  
oooooooooooo

C++ future  
oooo

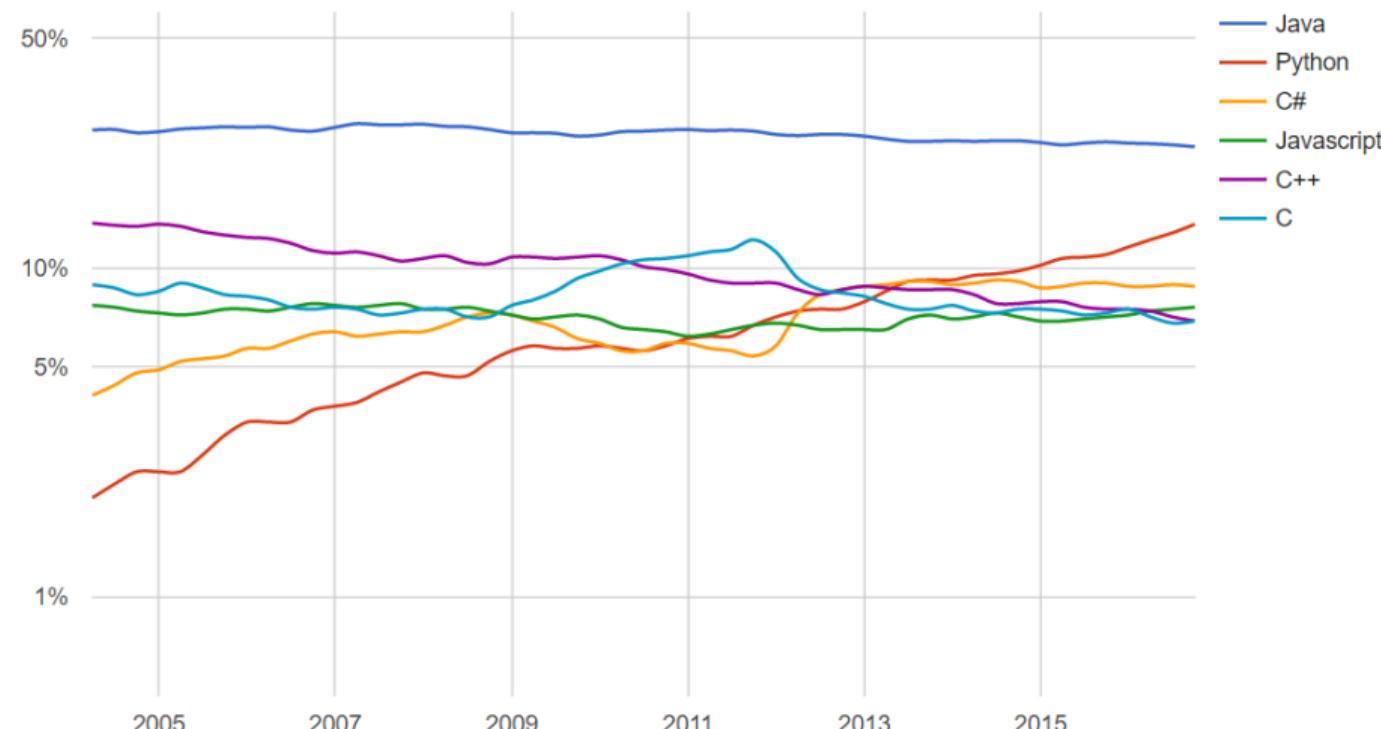
(R)evolution!  
ooooo

Language popularity  
ooooo●oooooo

Summary  
oooooo

# PYPL

## PYPL PopularitY of Programming Language



C with Classes  
oooooooo

Cfront era  
oooooooooooooooooooo

Standardization time  
oooooooooooo

C++ future  
oooo

(R)evolution!  
ooooo

Language popularity  
oooo●oooooooo

Summary  
oooooo

# PYPL

Worldwide, Oct 2016 compared to a year ago:

Rank	Change	Language	Share	Trend
1		Java	23.4 %	-0.8 %
2		Python	13.6 %	+2.5 %
3		PHP	9.9 %	-0.8 %
4		C#	8.8 %	-0.0 %
5	↑↑	Javascript	7.6 %	+0.6 %
6	↓	C++	6.9 %	-0.8 %
7	↓	C	6.9 %	-0.9 %
8		Objective-C	4.5 %	-0.7 %
9	↑	R	3.3 %	+0.7 %
10	↓	Swift	3.1 %	+0.3 %

C with Classes  
oooooooo

Cfront era  
oooooooooooooooooooo

Standardization time  
oooooooooooo

C++ future  
oooo

(R)evolution!  
ooooo

Language popularity  
oooo●oooo

Summary  
oooooo

# codeeval

codeeval MPCL

**"Most Popular Coding Languages"** is based on hundreds of thousands of data points we've collected by processing over 1,200,000+ **challenge submissions on codeeval.com** in (now) 26 different programming languages.  
Webpage: <http://blog.codeeval.com/>

C with Classes  
oooooooo

Cfront era  
oooooooooooooooooooo

Standardization time  
oooooooo

C++ future  
oooo

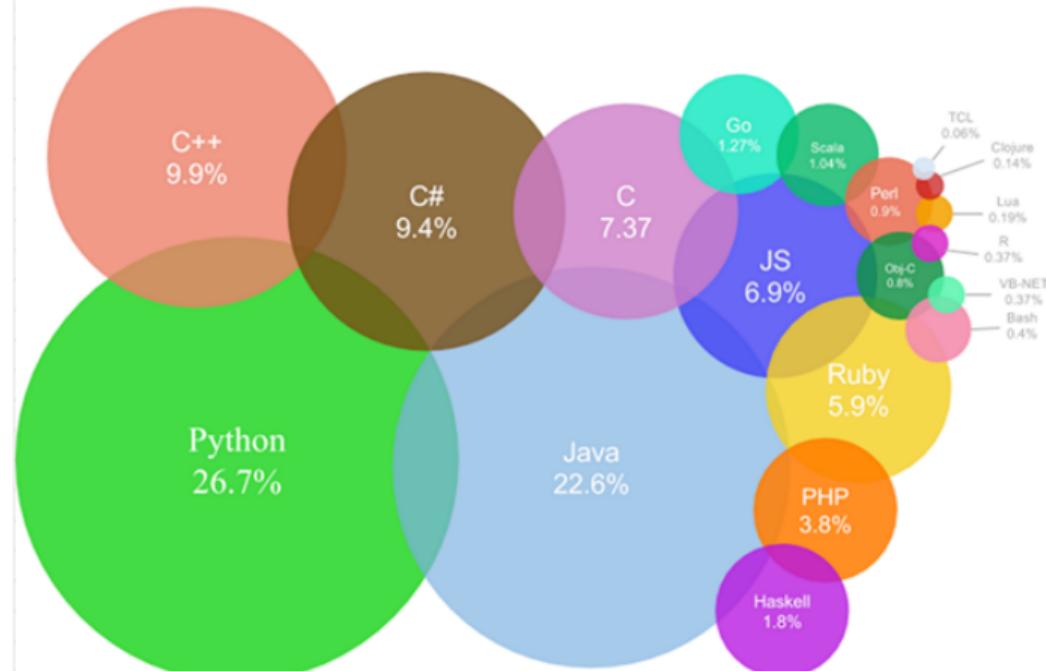
(R)evolution!  
ooooo

Language popularity  
ooooo●oooo

Summary  
oooooo

codeeval

## Most Popular Coding Languages of 2016



C with Classes  
oooooooo

Cfront era  
oooooooooooooooooooo

Standardization time  
oooooooo

C++ future  
oooo

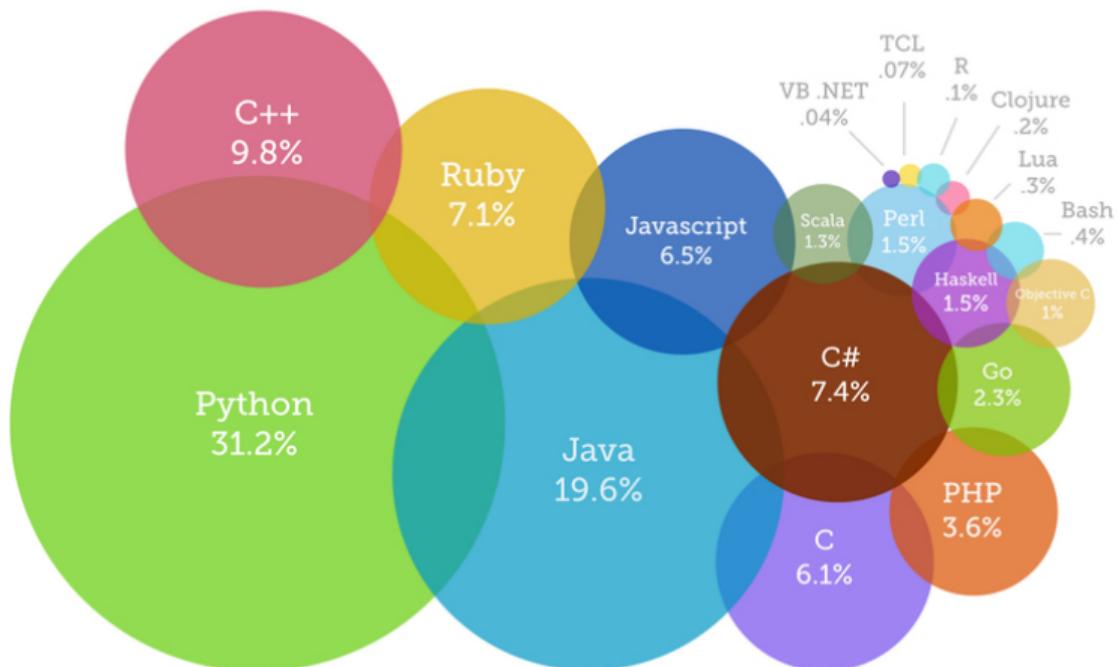
(R)evolution!  
ooooo

Language popularity  
ooooo●oooo

Summary  
oooooo

codeeval

## Most Popular Coding Languages of 2015



C with Classes  
oooooooo

Cfront era  
oooooooooooooooooooo

Standardization time  
oooooooooooo

C++ future  
oooo

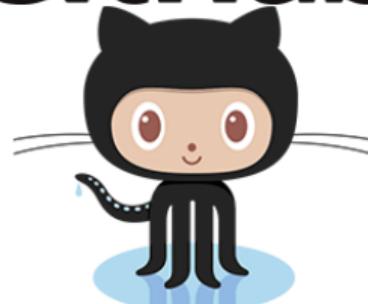
(R)evolution!  
ooooo

Language popularity  
oooooooo●oooo

Summary  
oooooo

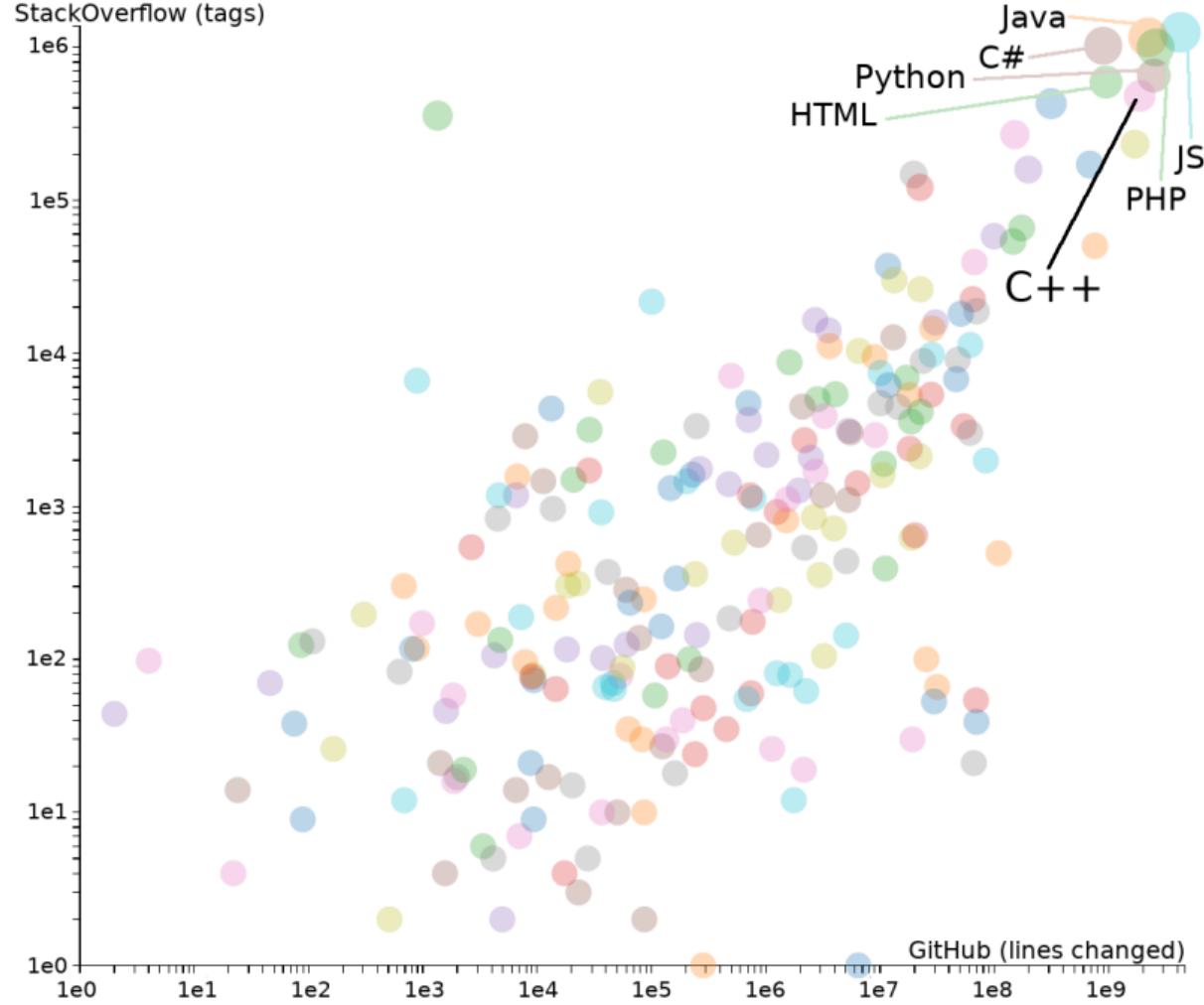
langpop.corger.nl

# GitHub



# stack overflow

Webpage: <http://langpop.corger.nl/>



C with Classes  
oooooooo

Cfront era  
oooooooooooooooooooo

Standardization time  
oooooooooooo

C++ future  
oooo

(R)evolution!  
ooooo

Language popularity  
oooooooo●ooo

Summary  
oooooo

# Stackoverflow issues

- good programmer == lazy programmer

C with Classes  
oooooooo

Cfront era  
oooooooooooooooooooo

Standardization time  
oooooooooooo

C++ future  
oooo

(R)evolution!  
ooooo

Language popularity  
oooooooo●ooo

Summary  
oooooo

# Stackoverflow issues

- good programmer == lazy programmer
- lazy programmer != good programmer

C with Classes  
oooooooo

Cfront era  
oooooooooooooooooooo

Standardization time  
oooooooooooo

C++ future  
oooo

(R)evolution!  
ooooo

Language popularity  
oooooooo●○○

Summary  
oooooo

# Stackoverflow issues

- good programmer == lazy programmer
- lazy programmer != good programmer
- good programmers do not reinvent the wheel

*Cutting corners to meet arbitrary management deadlines*



*Essential*

Copying and Pasting  
from Stack Overflow

O'REILLY®

*The Practical Developer*  
@ThePracticalDev

C with Classes  
oooooooo

Cfront era  
oooooooooooooooooooo

Standardization time  
oooooooooooo

C++ future  
oooo

(R)evolution!  
ooooo

Language popularity  
oooooooo●○○

Summary  
oooooo

# Stackoverflow issues

- good programmer == lazy programmer
- lazy programmer != good programmer
- good programmers do not reinvent the wheel
- do job once and never come back here

*Cutting corners to meet arbitrary management deadlines*



*Essential*

Copying and Pasting  
from Stack Overflow

O'REILLY®

*The Practical Developer*  
@ThePracticalDev

C with Classes  
oooooooo

Cfront era  
oooooooooooooooooooo

Standardization time  
oooooooooooo

C++ future  
oooo

(R)evolution!  
ooooo

Language popularity  
oooooooo●○

Summary  
oooooo

# C++ is not dead!

- In the worst case C++ is on 7th place
- In the best case C++ is on 3rd place
- C++ is one of the most popular languages!
- Next versions of C++ will be available soon



C with Classes  
oooooooo

Cfront era  
oooooooooooooooooooo

Standardization time  
oooooooooooo

C++ future  
oooo

(R)evolution!  
ooooo

Language popularity  
oooooooooooo●

Summary  
oooooo

# C++ is not dead!

Future C++:

- Type and resource safe



<sup>3</sup>Bjarne Stroustrup - *The Evolution of C++* - [https://www.youtube.com/watch?v=\\_wzc7a3Mc0s](https://www.youtube.com/watch?v=_wzc7a3Mc0s)

C with Classes  
oooooooo

Cfront era  
oooooooooooooooooooo

Standardization time  
oooooooooooo

C++ future  
oooo

(R)evolution!  
ooooo

Language popularity  
oooooooooooo●

Summary  
oooooo

# C++ is not dead!

Future C++:

- Type and resource safe
- Significantly simpler and clearer code



<sup>3</sup>Bjarne Stroustrup - *The Evolution of C++* - [https://www.youtube.com/watch?v=\\_wzc7a3Mc0s](https://www.youtube.com/watch?v=_wzc7a3Mc0s)

C with Classes  
oooooooo

Cfront era  
oooooooooooooooooooo

Standardization time  
oooooooooooo

C++ future  
oooo

(R)evolution!  
ooooo

Language popularity  
oooooooooooo●

Summary  
oooooo

# C++ is not dead!

## Future C++:

- Type and resource safe
- Significantly simpler and clearer code
- As fast or faster than anything else



<sup>3</sup>Bjarne Stroustrup - *The Evolution of C++* - [https://www.youtube.com/watch?v=\\_wzc7a3Mc0s](https://www.youtube.com/watch?v=_wzc7a3Mc0s)

C with Classes  
oooooooo

Cfront era  
oooooooooooooooooooo

Standardization time  
oooooooooooo

C++ future  
oooo

(R)evolution!  
ooooo

Language popularity  
oooooooooooo●

Summary  
oooooo

# C++ is not dead!

## Future C++:

- Type and resource safe
- Significantly simpler and clearer code
- As fast or faster than anything else
- Good at using "modern hardware" (more pipelines, more concurrency)



<sup>3</sup>Bjarne Stroustrup - *The Evolution of C++* - [https://www.youtube.com/watch?v=\\_wzc7a3Mc0s](https://www.youtube.com/watch?v=_wzc7a3Mc0s)

C with Classes  
oooooooo

Cfront era  
oooooooooooooooooooo

Standardization time  
oooooooooooo

C++ future  
oooo

(R)evolution!  
ooooo

Language popularity  
oooooooooooo●

Summary  
oooooo

# C++ is not dead!

## Future C++:

- Type and resource safe
- Significantly simpler and clearer code
- As fast or faster than anything else
- Good at using "modern hardware" (more pipelines, more concurrency)
- Significantly faster compilation catching many more errors



<sup>3</sup>Bjarne Stroustrup - *The Evolution of C++* - [https://www.youtube.com/watch?v=\\_wzc7a3Mc0s](https://www.youtube.com/watch?v=_wzc7a3Mc0s)

C with Classes  
oooooooo

Cfront era  
oooooooooooooooooooo

Standardization time  
oooooooooooo

C++ future  
oooo

(R)evolution!  
ooooo

Language popularity  
oooooooooooo

Summary  
oooooo

# Summary



C with Classes  
oooooooo

Cfront era  
oooooooooooooooooooo

Standardization time  
oooooooooooo

C++ future  
oooo

(R)evolution!  
ooooo

Language popularity  
oooooooooooo

Summary  
● ooooo

## Key messages

- ➊ C++ had a **clear aim**, which made it popular: to **organize code better without the loss of efficiency**

C with Classes  
oooooooo

Cfront era  
oooooooooooooooooooo

Standardization time  
oooooooooooo

C++ future  
oooo

(R)evolution!  
ooooo

Language popularity  
oooooooooooo

Summary  
●ooooo

## Key messages

- ➊ C++ had a **clear aim**, which made it popular: to **organize code better without the loss of efficiency**
  
- ➋ C++ is even more popular now, because of new standards: **C++11** and **C++14**

C with Classes  
oooooooo

Cfront era  
oooooooooooooooooooo

Standardization time  
oooooooo

C++ future  
oooo

(R)evolution!  
ooooo

Language popularity  
oooooooooooo

Summary  
●ooooo

## Key messages

- ➊ C++ had a **clear aim**, which made it popular: to **organize code better without the loss of efficiency**
- ➋ C++ is even more popular now, because of new standards: **C++11** and **C++14**
- ➌ In future C++ will be **one of the most popular programming languages** so it's worth learning

C with Classes  
oooooooo

Cfront era  
oooooooooooooooooooo

Standardization time  
oooooooooooo

C++ future  
oooo

(R)evolution!  
ooooo

Language popularity  
oooooooooooo

Summary  
○●oooo

# Worth reading / watching

 Bjarne Stroustrup – *The Evolution of C++ - Past, Present, and Future*  
CppCon 2016

[https://www.youtube.com/watch?v=\\_wzc7a3Mc0s](https://www.youtube.com/watch?v=_wzc7a3Mc0s)

 Herb Sutter – *One C++*  
Going Native 2013

<https://channel9.msdn.com/Events/GoingNative/2013/Keynote-Herb-Sutter-One-Cpp>

 Kate Gregory – *Stop teaching C*  
CppCon 2015

<https://www.youtube.com/watch?v=YnWhqhNdYyk>

 Bjarne Stroustrup – *A History of C++: 1979-1991*  
<http://www.stroustrup.com/hopl2.pdf>

 Scott Meyers – *Effective Modern C++*

C with Classes  
oooooooo

Cfront era  
oooooooooooooooooooo

Standardization time  
oooooooooooo

C++ future  
oooo

(R)evolution!  
ooooo

Language popularity  
oooooooooooo

Summary  
○○●○○

# How many book covers are in my presentation?

C with Classes  
oooooooo

Cfront era  
ooooooooooooooo

Standardization time  
oooooooo

C++ future  
oooo

(R)evolution!  
oooo

Language popularity  
oooooooooooo

Summary  
○○●○○

# How many book covers are in my presentation?



C with Classes  
oooooooo

Cfront era  
oooooooooooooooooooo

Standardization time  
oooooooooooo

C++ future  
oooo

(R)evolution!  
ooooo

Language popularity  
oooooooooooo

Summary  
ooo●oo

# What now?

## What now?

- Look forward to C++17 and learn it

C with Classes  
oooooooo

Cfront era  
oooooooooooooooooooo

Standardization time  
oooooooooooo

C++ future  
oooo

(R)evolution!  
ooooo

Language popularity  
oooooooooooo

Summary  
ooo●oo

# What now?

## What now?

- Look forward to C++17 and learn it
- Watch a lot of videos from C++ conferences

C with Classes  
oooooooo

Cfront era  
oooooooooooooooooooo

Standardization time  
oooooooooooo

C++ future  
oooo

(R)evolution!  
ooooo

Language popularity  
oooooooooooo

Summary  
ooo●oo

# What now?

## What now?

- Look forward to C++17 and learn it
- Watch a lot of videos from C++ conferences
- Visit [isocpp.org](http://isocpp.org)

C with Classes  
oooooooo

Cfront era  
oooooooooooooooooooo

Standardization time  
oooooooooooo

C++ future  
oooo

(R)evolution!  
ooooo

Language popularity  
oooooooooooo

Summary  
oooo●○

## Famous quotation

*“Learn C++. It’s an investment.”*

C with Classes  
oooooooo

Cfront era  
oooooooooooooooooooo

Standardization time  
oooooooooooo

C++ future  
oooo

(R)evolution!  
ooooo

Language popularity  
oooooooooooo

Summary  
oooo●○

## Famous quotation

*“Learn C++. It's an investment.”*

— Łukasz Ziobroń





KEEP  
CALM  
AND  
~~DON'T~~ ASK  
QUESTIONS