

Introduction

Object-Oriented Programming with C++
1st Lecture, Mar. 26, 2020

Instructor:
Amiran Malania

Game Plan

- ❑ Welcome
- ❑ Logistics
- ❑ History and philosophy of C++
- ❑ Practice: C++ basics

Game Plan

- ❑ **Welcome**
- ❑ Logistics
- ❑ History and philosophy of C++
- ❑ Practice: C++ basics

Instructor

- ❑ BSc in Computer Science from *San Diego State University*
- ❑ Software Engineer at *Qarva*
- ❑ Interested in Computer Systems and Quantum Computing
- ❑ Amateur runner

Game Plan

- ❑ Welcome
- ❑ **Logistics**
- ❑ History and philosophy of C++
- ❑ Practice: C++ basics

Expectations and requirements

- ❑ Basics of programming(data types, functions, arrays, structs, pointers)
- ❑ Ability to independently set up working infrastructure(IDE/Editor, compiler)
- ❑ Use of google cloud for sharing documents

Syllabus(pdf)

How to succeed

- ❑ Don't miss classes
- ❑ Ask questions!!!
- ❑ Write assignments

Game Plan

- ❑ Welcome
- ❑ Logistics
- ❑ **History and philosophy of C++**
- ❑ Practice: C++ basics

History and philosophy of C++

“If you wish to make an apple pie from scratch, you must first invent the universe”

- Carl Sagan



History and philosophy of C++

“If you wish to *understand C++* from scratch, you must first invent the universe”

- (not) Carl Sagan



C++ History: Assembly

```
section      .text
global      _start                ;must be declared for linker (ld)

_start:                                           ;tell linker entry point

    mov     edx,len                ;message length
    mov     ecx,msg                ;message to write
    mov     ebx,1                  ;file descriptor (stdout)
    mov     eax,4                  ;system call number (sys_write)
    int     0x80                  ;call kernel
    mov     eax,1                  ;system call number (sys_exit)
    int     0x80                  ;call kernel

section      .data
msg          db  'Hello, world!',0xa             ;our dear string
len          equ $ - msg                        ;length of our dear string
```

C++ History: Assembly

Some **pros** features of Assembly

- ❑ Unbelievably simple instructions (move bits around, add, subtract)
- ❑ Well written assembly is **extremely** fast.
- ❑ Gives you complete control over your program.

So, why don't we code in Assembly?

C++ History: Assembly

```
section      .text
global      _start                ;must be declared for linker (ld)

_start:                                           ;tell linker entry point

    mov     edx,len                ;message length
    mov     ecx,msg                ;message to write
    mov     ebx,1                  ;file descriptor (stdout)
    mov     eax,4                  ;system call number (sys_write)
    int     0x80                  ;call kernel
    mov     eax,1                  ;system call number (sys_exit)
    int     0x80                  ;call kernel

section      .data
msg          db  'Hello, world!',0xa             ;our dear string
len          equ $ - msg                        ;length of our dear string
```

C++ History: Moving Forward

Writing assembly was too difficult but computers only understood assembly

C++ History: Moving Forward

Writing assembly was too difficult but computers only understood assembly

Idea:

- ❑ Source code can be written in a more intuitive language
- ❑ An additional program can convert it into assembly

C++ History: Moving Forward

Writing assembly was too difficult but computers only understood assembly

Idea:

- ❑ Source code can be written in a more intuitive language
- ❑ An additional program can convert it into assembly



This is called **compiler**

C++ History: Invention of C

Dennis Ritchie and Ken Thompson
create C programming language in
1972

C made it easy to write code that was

- ❑ Fast
- ❑ Simple
- ❑ Cross-platform



C++ History: Invention of C

C was popular since it was simple.

This was also its weakness:

- ❑ No **objects** or **classes**
- ❑ Difficult to write code that worked **generically**
- ❑ Tedious when writing large programs

C++ History: Bjarne Stroustrup

In 1983, the first vestiges of C++ were created by Bjarne Stroustrup.

He wanted a language that was:

- ❑ Fast
- ❑ Simple to use
- ❑ Cross-platform
- ❑ Had high-level features



C++ History: Bjarne Stroustrup

In 1983, the first vestiges of C++ were created by Bjarne Stroustrup.

He wanted a language that was:

- ☐ Fast
- ☐ Simple to use
- ☐ Cross-platform
- ☐ Had high-level features



C++ History: Bjarne Stroustrup

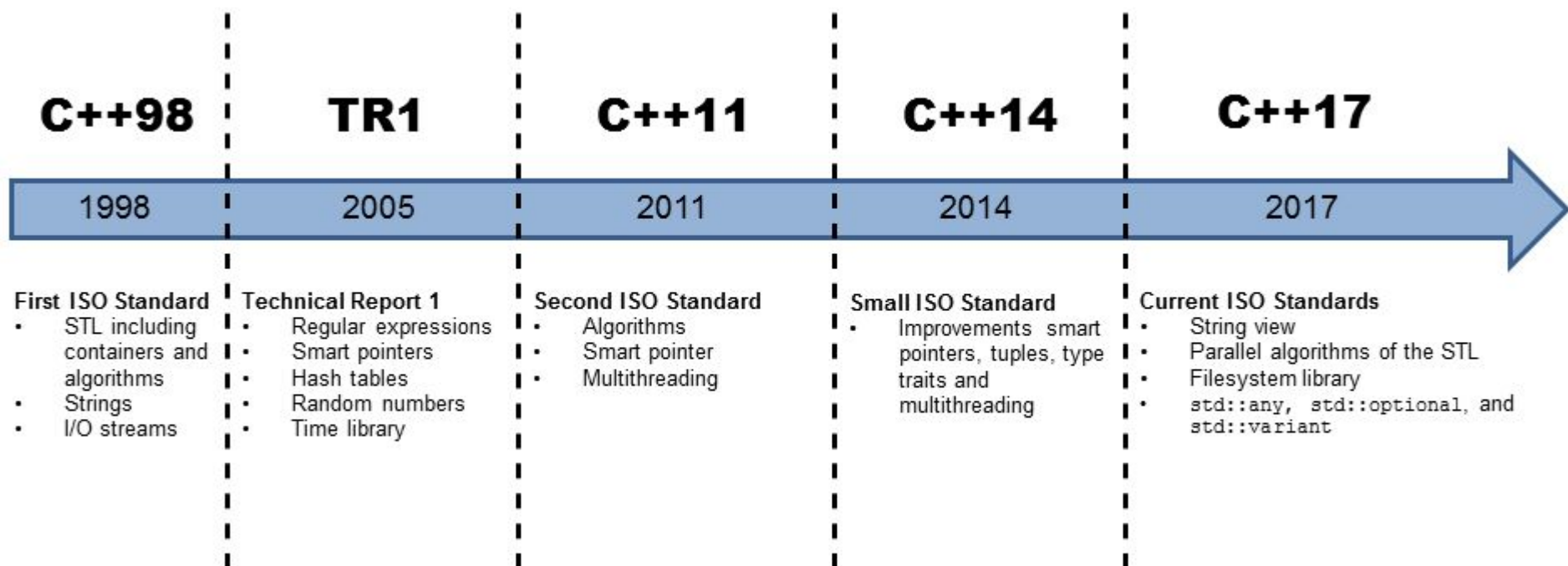
In 1983, the first vestiges of C++ were created by Bjarne Stroustrup.

He wanted a language that was:

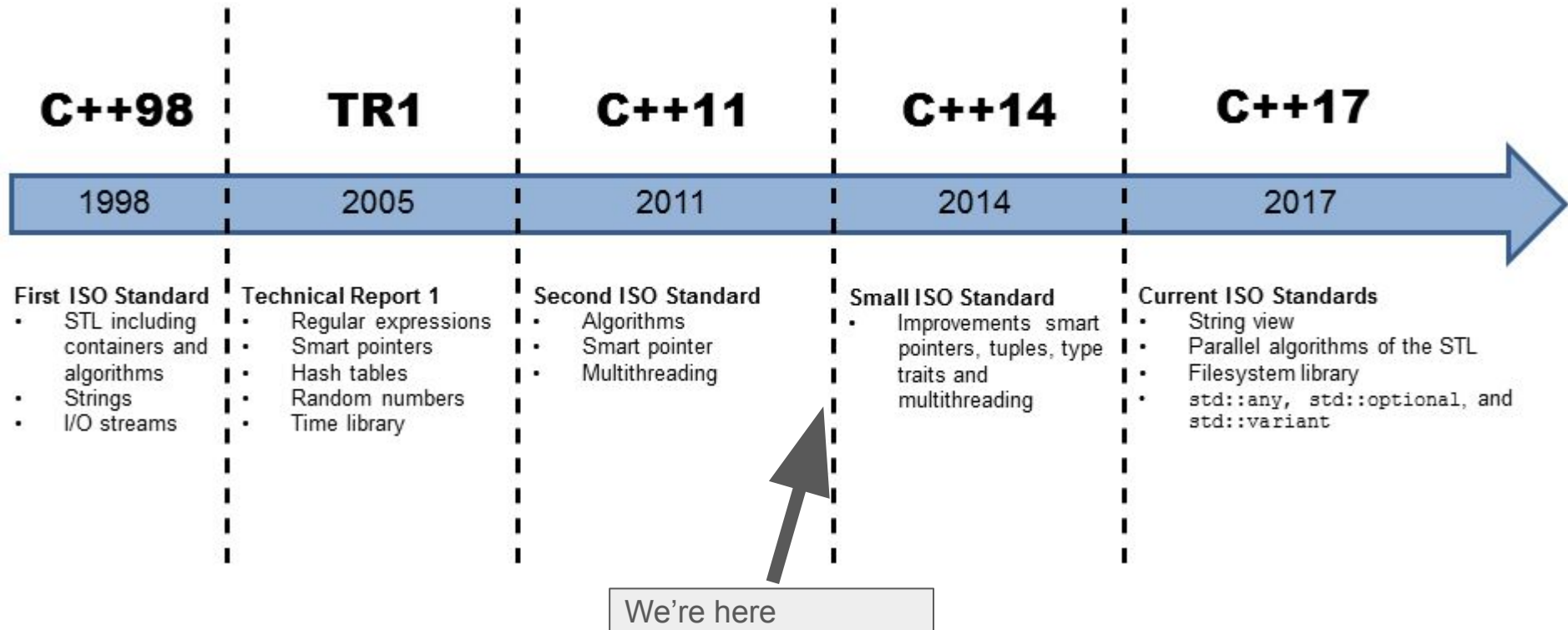
- ☐ Fast
- ☐ Simple to use
- ☐ Cross-platform
- ☒ Had high-level features



C++ History: Evolution of C++



C++ History: Evolution of C++



Game Plan

- ❑ **Welcome**
- ❑ Logistics
- ❑ History and philosophy of C++
- ❑ **Practice: C++ basics**

End