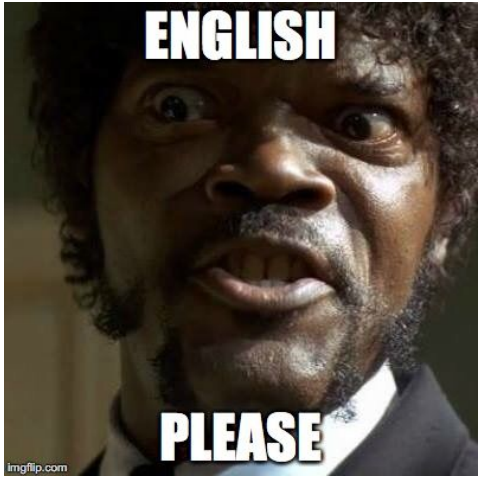
System Programming with C++

Course organization

Languages

Languages

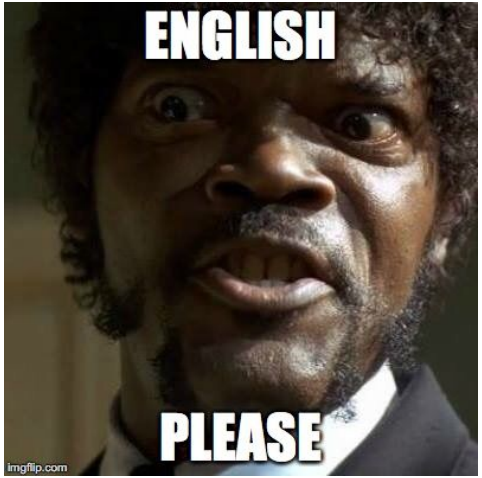
Only C++ and English



Languages

Only C++ and English

We do not use Russian here.



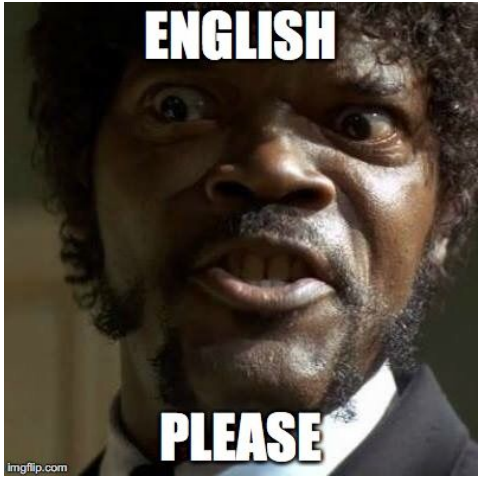
Languages

Only C++ and English

We do not use Russian here.

- Lectures,
- Discussions,
- Problems descriptions,
- Consultations,
- Exams!

…all in English.

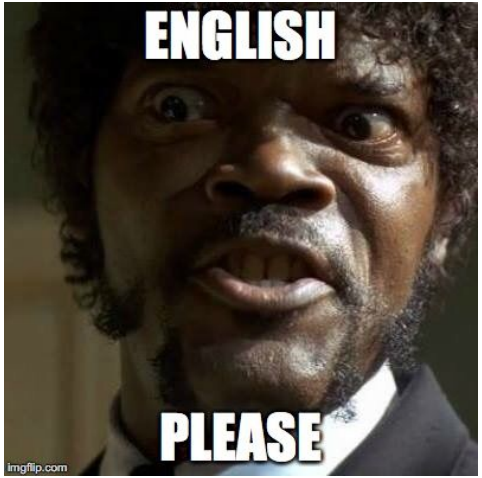


Languages

Only C++ and English

We do not use Russian here.

- Lectures,
- Discussions,
- Problems descriptions,
- Consultations,
- Exams!



…all in English. Second retake can be in 🇷🇺

Teacher

Ivan Ugliansky

@ugliansky

ivan.ugliansky@gmail.com

+7-913-763-3346

And your Telegram group!

What this course is about?

What this course is about?

- C++ language itself

What this course is about?

- C++ language itself
  - Main features
    - ✓ classes, references, move semantics, etc
    - ✓ inheritance, virtual methods
    - ✓ templates and concepts
    - ✓ exceptions
    - ✓ std/stl, smart pointers, threads

What this course is about?

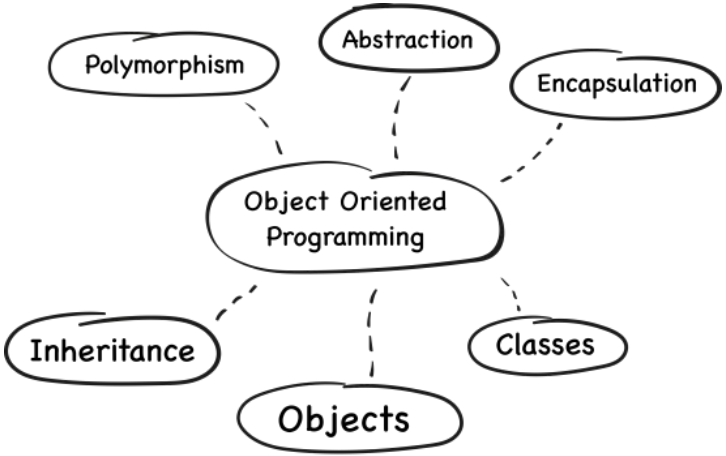
- C++ language itself
  - Main features
  - Their implementation

What this course is about?

- C++ language itself
  - Main features
  - Their implementation
    - ✓ How do virtual calls actually work?
    - ✓ What is the price of templates?
    - ✓ Which optimizations does compiler make?

What this course is about?

- C++ language itself
- Conceptions of Object Oriented Programming



What this course is about?

- C++ language itself
- Conceptions of Object Oriented Programming

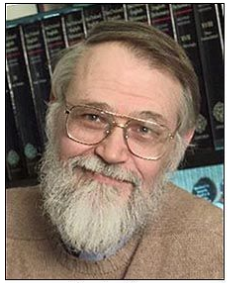
Preconditions for this course:

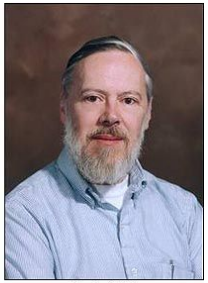
What this course is about?

- C++ language itself
- Conceptions of Object Oriented Programming

Preconditions for this course:

- C language: pointers and structs
  (macroses as well)


Brian Kernighan


Dennis Ritchie

What this course is about?

- C++ language itself
- Conceptions of Object Oriented Programming

Preconditions for this course:

- C language: pointers and structs
  (macroses as well)
- Java/Python and some basic knowledge
  about classes and their methods

Why learn C++?

Why learn C++?

- Still the main tool in system programming

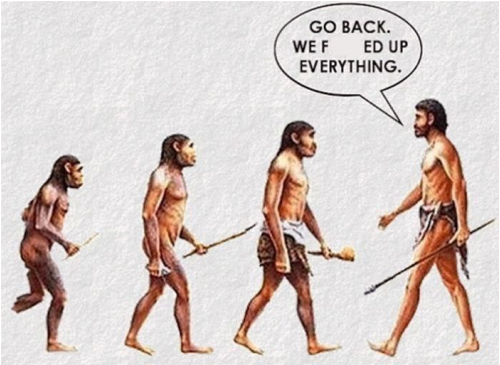
Why learn C++?

- Still the main tool in system programming (but C is still strong and Rust is new wave)

Why learn C++?

- Still the main tool in system programming (but C is still strong and Rust is new wave)

- Brutal evolution of the language



Why learn C++?

- Still the main tool in system programming (but C is still strong and Rust is new wave)
- Brutal evolution of the language
- Not the worst example of OOP paradigm

Why learn C++?

- Still the main tool in system programming (but C is still strong and Rust is new wave)
- Brutal evolution of the language
- Not the worst example of OOP paradigm
- 

  Because Losing Is Fun

How will we learn C++?

How will we learn C++?

- 16 lectures

How will we learn C++?

- 16 lectures + X consultations

How will we learn C++?

- 16 lectures + X consultations
- Practice = "tiny tasks"

How will we learn C++?

- 16 lectures + X consultations
- Practice = "tiny tasks", but quite unusual:

  1. Not so tiny.

How will we learn C++?

- 16 lectures + X consultations
- Practice = "tiny tasks", but quite unusual:
  1. Not so tiny.
  2. Next task can be based on previous one.

How will we learn C++?

- 16 lectures + X consultations
- Practice = "tiny tasks", but quite unusual:

  1. Not so tiny.
  2. Next task can be based on previous one.
  3. Tests and sanitizers!

How will we learn C++?

- 16 lectures + X consultations
- Practice = "tiny tasks" + "coursework":
    1. Quite huge project!

How will we learn C++?

- 16 lectures + X consultations
- Practice = "tiny tasks" + "coursework":
  1. Quite huge project! Made by a team of 2-3 students.

How will we learn C++?

- 16 lectures + X consultations
- Practice = "tiny tasks" + "coursework":

  1. Quite huge project! Made by a team of 2-3 students.
  2. Some topics will be suggested, but feel free to suggest something new. Be creative!

How will we learn C++?

- 16 lectures + X consultations
- Practice = "tiny tasks" + "coursework":

  1. Quite huge project! Made by a team of 2-3 students.
  2. Some topics will be suggested.
  3. 4 milestones during the semester.

How will we learn C++?

- 16 lectures + X consultations
- Practice = "tiny tasks" + "coursework":

04.03.24

```
Project topic approved;
Team formed; Roles chosen;
Further plan prepared;
```

How will we learn C++?

- 16 lectures + X consultations
- Practice = "tiny tasks" + "coursework":

04.03.24

Project topic approved;
Team formed; Roles chosen;
Further plan prepared;

01.04.24

"Proof of concept"
prototype is ready;
Initial review passed;

How will we learn C++?

- 16 lectures + X consultations
- Practice = "tiny tasks" + "coursework":

04.03.24

Project topic approved;
Team formed; Roles chosen;
Further plan prepared;

01.04.24

"Proof of concept"
prototype is ready;
Initial review passed;

29.04.24

Main functionality
implemented;
Second review passed;

How will we learn C++?

- 16 lectures + X consultations
- Practice = "tiny tasks" + "coursework":

04.03.24

Project topic approved;
Team formed; Roles chosen;
Further plan prepared;

01.04.24

"Proof of concept" prototype is ready;
Initial review passed;

29.04.24

Main functionality implemented;
Second review passed;

27.05.24

Polishing finished;

Presentation of the project.

Points system

Points system

- ✓ Tiny tasks — [0; 20] points
- ✓ Coursework — [0; 40] points

Points system

- ✓ Tiny tasks — [0; 20] points
- ✓ Coursework — [0; 40] points

  - Passing milestones can give you 5/10/10/15 points each.

  - Missing a deadline give you 50% penalty.

Points system

- ✓ Tiny tasks — [0; 20] points
- ✓ Coursework — [0; 40] points
- ✓ Exam — [0; 40] points

Points system

- ✓ Tiny tasks — [0; 20] points
- ✓ Coursework — [0; 40] points
- ✓ Exam — [0; 40] points

- ✓ [85; 100] points — A (5)
- ✓ [70; 85) points — B (4)
- ✓ [55; 70) points — C (3)

Points system

- ✓ Tiny tasks — [0; 20] points
- ✓ Coursework — [0; 40] points
- ✓ Exam — [0; 40] points

- ✓ [85; 100] points — A (5)
- ✓ [70; 85) points — B (4)
- ✓ [55; 70) points — C (3)

60 points for the practice?

Ok, take 10 more free points!

Materials and alternative courses

1. MIPT Course by Konstantin Vladimirov

Materials and alternative courses

1. MIPT Course by Konstantin Vladimirov

2. C++ course from Computer Science Center (Part 1, Part 2)

Materials and alternative courses

1. MIPT Course by Konstantin Vladimirov

2. C++ course from Computer Science Center (Part 1, Part 2)

3. Books?

Materials and alternative courses

1. MIPT Course by Konstantin Vladimirov

2. C++ course from Computer Science Center (Part 1, Part 2)

3. Books? Stroustrup? Alexandrescu?

Materials and alternative courses

1. MIPT Course by Konstantin Vladimirov

2. C++ course from Computer Science Center (Part 1, Part 2)

3. Books? Stroustrup? Alexandrescu? Sure, if you want.

Finally, we are ready for action!