++it Italian C++ Community

www.italiancpp.org

The dog ate my slides

Actually I don’t really have a dog.

Who am I?

Codemachine

- C++03,11,14 Architect and Developer

European Community – Joint Research Center
Institute for transuranium elements

- C++11 & QT Developer

Hobby

- C++ developer

Blog

- C++ explained to my dog http://marcofoco.com
  - (I assure I don’t have a dog – really)

Teaching C++14 from scratch, on Raspberry PI

Why?

- Why not?

Where?

- Spazio YATTA, Milan (Maker Space)

Seriously, why Raspberry PI and why C++14?

- Raspberry: Appealing for the target
- C++14: Because it’s easier!

Goals

For the community/audience

- Teaching C++14 from scratch
- Configuring a Raspberry PI for C++14 development machine
- Giving feedback of this experiment to you (this presentation!)

Personal

- Getting back to teaching
- Forcing myself to experiment
- Having fun!

Structure

4 evenings, 3 hour per encounter

1. Raspberry & Setup for C++14
   History of C++
   Language constructs
   Types and Variables
2. Functions and Lambda
   References. Copying and moving
3. Standard Library
   Experimenting with algorithms
4. Using external libraries
   Final project

Audience

8 people (average presence 7-8)

heterogeneous experience

- C++
- C on Arduino
- C, university student level (3x)
- Matlab
- No experience at all (2x)

Spontaneous meetings

- Students started self-organizing and meeting together to produce the final project

Results

Final project

- Activating a relay when a face was detected on the camera
  - Used OpenCV for Face Detection
  - Used WiringPI library for driving Inputs and Outputs

Considerations

- 12 hours are not enough.
- Night is the right time for coding, not for studying.

C++14: A simplified C++

Auto

No more pointers

Cleaner syntax

Comprehensive library

auto

Isn‘t auto a complex topic?

- No, you just need to explain the concept of “type associated to an expression”

How auto simplifies things?

- Less types to explain
  - Iterators
  - Lambdas

You did explain lambdas?!?

- Yes. Having explained functions and type deduction, even generic lambdas are not a problem (when you capture-by-copy)

No more (raw) pointers

Maximize the use of local variables

Parameter passing by-value, const &, &&

- Wait! Isn‘t move semantics a difficult topic?
- The most difficult part is to explain why, after a move, the compiler don't warn you if you use the variable again!

Smart pointers

Construction with make_shared, make_unique constructor-forwarding functions

Cleaner syntax

Good

- Range-based for
- Returning complex objects by value

Some complexities

- You still need to explain const-correctness
- Parameter passing is confusing By-value, const&, &&, (&)

Comprehensive library

Containers, strings, and so on… (all the classes from C++03)

- No news here!

String literals are your friend!

- Literals helps to deduce the right type, and use the right operators for strings
- Issues with people who did study C or C++ before

Completely new set of errors! (1)

Classic memory management in C/C++

- Memory allocation concepts
- Memory leaks
- Double deallocations
- ...

Shared pointers in C++14

- Circular references
- Unreadable errors (wrong arguments on a simple constructor will generate 70 lines of errors!
  - (hackish solution: ???)

Completely new set of errors! (2)

Forgetting about a trailing “s” on strings

- Sometimes works as expected! (not that good, actually, but ok)
- Sometimes prevent things from working (very good)
- Sometimes changes the meaning of the code (bad)
- Sometimes it makes bad code to compile and produce a completely unexpected result (very-very-very bad)

Conclusions

Unexpected results!

- Self organization and use of OpenCV came from the students

Too early

- Error messages are a big obstacle: We need CONCEPTS!

Next experiment will be C++17 ☺

Questions

?