

Geant4 & NEXUS Hands-on Tutorial

Justo Martín-Albo (IFIC)

IFIC • 24–25 February 2020

GOAL AND MEANS

Our main goal for this tutorial:

- Become familiar enough with NEXUS to be able to start contributing to its development.

How?

- Review the C++ concepts most frequently used in Geant4 and NEXUS.
- Introduce the basic components that form a Geant4 simulation.
- Learn how to build and run NEXUS, and review its structure and organization.
- Show what documentation resources can be useful when writing NEXUS code.
- Write a full simulation within NEXUS and analyze the generated data.

NEXUS (*NEXt Utility for Simulation*) is the Geant4-based detector simulation of the NEXT experiment. It handles the different *detector geometries* (DEMO, NEW, NEXT-100...) and *event generators* (DBDs, radioactive backgrounds, muons...) we need, and produces output files (using now HDF5) in a common format understood by the software downstream.

Geant4 is a C++ library for the simulation of the passage of particles through matter using Monte-Carlo methods.

**(A very brief
introduction to C++)**

Fork and clone the following repository:

<https://github.com/jmalbos/nexus-tutorial>

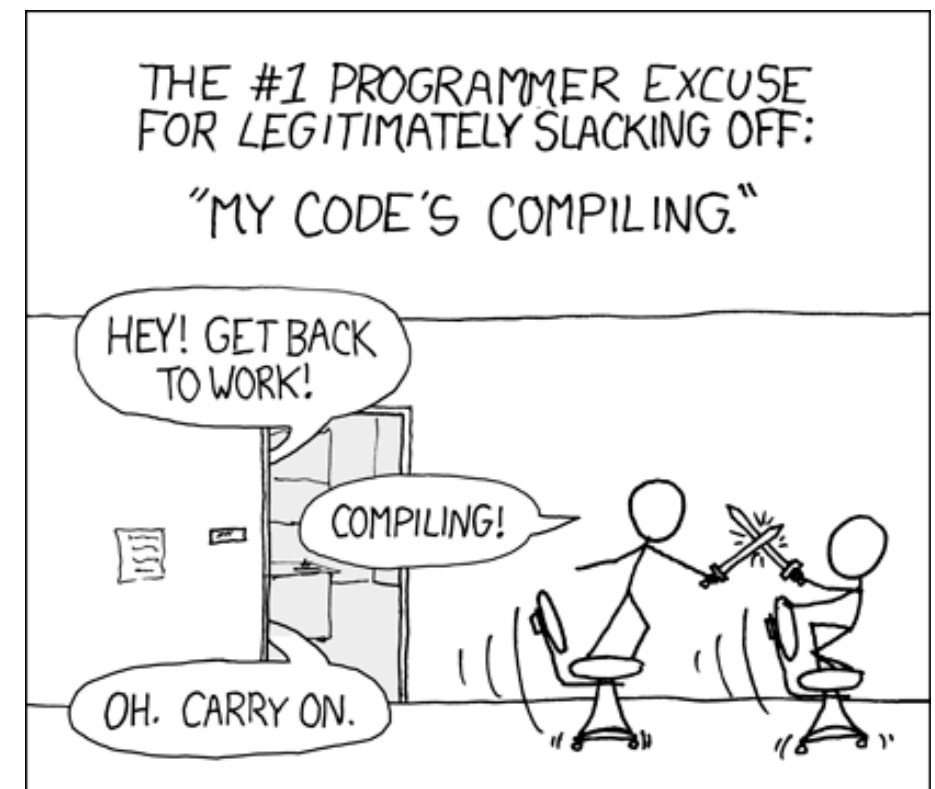
Build the examples of the C++ introduction:

```
cd nexus-tutorial/C++Intro
```

```
mkdir build && cd $_;
```

```
cmake ..
```

```
cmake -build .
```



- 0.** Function declaration and definition. Main function and control flow. Scope of variables and namespaces.
- 1.** Passing arguments by value, reference and pointer.
- 2.** Declaration and definition of classes. Constructors and destructors.
- 3.** Class inheritance. Memory allocation.
- 4.** Polymorphism. Vectors.

SPECIFIER	WITHIN SAME CLASS	WITHIN DERIVED CLASS	FROM OUTSIDE
public	✓	✓	✓
protected	✓	✓	✗
private	✓	✗	✗