FCC – C++ Coding Conventions?

So then I typed GOTO 500—and here I am!

Zbyněk Drásal
(CERN)

Coding Conventions - General

- Keep the code as simple & clear as possible:
  - Use inheritance and templates with care
  - Consider carefully whether aggregation or composition is needed
  - Avoid multiple inheritance or friends if not necessary
  - Avoid side-effects of your coding, if side-effects are required, document!
  - Document the code properly (use e.g. Doxygen to gen. Documentation)
  - Use code formatter (e.g. `astyle`) before pushing the code to repository, the diff will then always provide a valid information!

→ **the code should compile without warnings!!!**

Coding Conventions - Names

- Variables, functions & object names → use intuitive English names
  - Class and Type names start with a capital letter: `GenDetector`
  - Enum types start with E followed by a capital letter: `EdetTypes`
  - Local variable and Method names start with a lower case letter: `detName, simulateEvent`
    - With prefix m_ for member variables: `m_detName`
    - With prefix s_ for static member variables: `s_detCounter`
    - With prefix g_ for static const variables (global constants): `g_softName`
    - With prefix c_ for non-global constants (including enums): `c_detSize`
  - Preprocessor definitions are all in capital letters with underscore between names: `DET_ID`
- Forbidden:
  - Avoid abbreviations: use `PixelDetector` instead of `Pdet`
- Other:
  - Use get/set names for getter, setter methods: `getEnergy()`
  - Use plural for list, vector … : `std::vector<MCParticle> mcParticles` instead of `mcParticleVec`

Coding Conventions - Programming

- Programming tips:
  - Use `const.` whenever possible → will inform about unwanted changes of variables' value already during the compilation time
  - Initialize variables immediately in the same statement: `float x=0.0;` for classes in the constructor or header file (C++11)
  - Pass input parameters by value: const reference or const pointer, avoid using pointers where not necessary
  - Forbidden: Namespace is forbidden to be declared in header files & header files must be included outside of namespace region → avoid recursive declarations!!!
- Files structure:
  - Name header files `*.h`, source files `*.cc`
  - Use forward declaration of `class` in header file, instead of `#include`
  - In header file use the following to avoid multiple including!!!:

```
#ifndef FILENAME_H // first line
#define FILENAME_H ...
#ENDIF FILENAME_H // last line
```

Coding Conventions – Class Structure

- Class structure tips:
  - Include files via `#include`
  - Define preprocessor variables via `#DEFINE`
  - Define public, protected and private sections in this order! Each section will contain:
    - typedefs and enums
    - constructors and destructor
    - operators
    - other methods
    - data members
- Documentation tips:
  - Describe class properties & its aim
  - Each method should be commented (if its functionality not obvious from its name) + the inputs & outputs must be commented
  - Each variable should be commented if functionality not obvious: e.g. `particleEnergy`

Coding Conventions – Disclaimer

- Any suggestions for a disclaimer written in a header file?

```
/*****************************************************
 * FCCSW (FCC Software framework based on GAUDI)      *
 *                                                    *
 * Copyright(C) YEAR  FCC Collaboration (CERN)        *
 *                                                    *
 *                                                    *
 * Author: The FCC Collaboration (CERN)               *
 *  Contributors: YOUR NAME                           *
 *  Repository: https://github.com/HEP-FCC/           *
 *                                                    *
 * This software is provided "under license"...       *
 *                                                    *
 *****************************************************/
```