Computing sessions 2019: assessment skill list

Skill category	Minimum	Satisfying	Very satisfying
1. Knowing C-programming basics	• Writing a “Hello World!” program • Asking questions to the user • Writing functions		
2. Using the standard library	• Using std::cout, std::string, std::fstream	• Using std::vector, std::stringstream and cmath.	• Using algorithms, iterators and manipulators.
3. Writing a C++ class	• Writing a simple class with: constructor without and with arguments, destructor, mutators, accessors and “print” function. • Instantiating and testing the implemented class.	• The class contains all the functionalities required by the specifications.	• Implementing operator overloading and copy constructor. • Using properly the reserved keywords “const” and “static”.
4. Coding algorithms	• Algorithms work and give the correct results.	• The code is robust: it is protected against bad inputs. • Managing properly the dynamic memory allocation (delete).	• The code is efficient: efforts are achieved for saving time.

Computing sessions 2019: assessment skill list

5. Using ROOT functionalities	• Plotting 1D and 2D histograms. • Using the C++ interactive interpreter of ROOT.	• Saving data in ROOT files. • Fitting data with a predefined function.	• Getting parameters of the fit.
6. Building a program	• Compiling and linking a simple program. • Reading compilator messages and fixing the code. • Providing to the supervisors a compilable program.	• Compiling a project based on several source files. • Compiling with external libraries (especially ROOT)	• Using a Makefile for building a project.
7. Documenting and preserving the code	• The source files are organized in folders. • One file for each class. • Saving the code on a USB stick	• Documenting the code by putting comments inside the source files: header for the file, header of the functions, explanations in algorithm code • Following the same code conventions in the same project.	• Writing a README and INSTALLATION files for explaining the goal of the program and how to compile it. • Generating Doxygen documentation related to the code.
8. Solving an instrumentation problem	• Acquiring data and saving them in a file.	• Analyzing data: histograms, common statistical information, correlation study	• Combining existed classes for solving a problem. • Calibrating sensors.

Computing sessions 2019: assessment skill list

- Je pars du principe que l'on n'évalue pas séance par séance mais sur l'ensemble des séances.
- Pour valider le module, il faut au moins que toutes les « minimum » sont remplies et la moitié des cases « satisfying ».
- Traduction en note sur 20 :
    - Valider simplement le module = 10
    - Tout = 20. Autre : interpolation linéaire (10 points pour 12 niveaux de compétences).
    - Rien = 0. Autre difficile … car compétences min demandées.