|  |  |  |  |
| --- | --- | --- | --- |
| 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. |
| **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) | * Linux/MacOSX: using a Makefile for building a project. * Windows: using a Visual Studio solution for dealing with 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 git repository. | * Documenting the code by putting comments (inside the source files: header for the file, ....) * Commenting properly each git commit. * 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. |