

# Coding Standards

- All programs are to have a make file that will compile the program to a g++ executable which runs on Debian based Linux systems, i.e. the HPCL lab computers.
- AStyle alignment is correct. I will accept either ANSI style alignment or Java style alignment.
- No global variables.
- All variables must be initialized before they are used.
- Constants are to be all uppercase.
- Prefer using return values over output parameters: they improve readability, and often provide the same or better performance. If output-only parameters are used, they should appear after input parameters.
- Prefer small and focused functions.
- Default arguments are allowed when the default is guaranteed to always have the same value.
- All functions are placed below the main with prototypes at the top between the namespace and the beginning of the main.
- All header files are to be guarded.
- For non-templated class structures, there are to be separate declaration (header .h) and implementation (.cpp) files.
- For templated classes, the entire class code is to be in a guarded header file (.h). The specification for the class is to be at the top and all implementation code below that.
- No inline code in the specification exceeding 2 lines in Astyle format.
- All commented code (i.e. not in use by the program) is to be removed.
- Commenting is to follow the commenting requirements in the Commenting Standards page.