Programming Standards

When writing programs in a organization such as a business or open-source project, there are typically coding standards which must be followed. For example, Google's C++ coding standards can be found at [https://google.github.io/styleguide/cppguide.html (Links to an external site.)Links to an external site.](https://google.github.io/styleguide/cppguide.html).

In this course, code submitted for labs and programming assignments should follow the minimal standards listed below (much simpler than the typical organization's standards). Failure to follow these standards may result in points deducted (NOTE: these standards may be updated during the course, but will not be applied retroactively to assignments before the updates).

* Assignments must be submitted as NetBeans project exports (File -> Export Project -> To ZIP...).
* Programs must conform to the C++14 standard.
* The main program must be in a file called main.cpp.
* The definition for each outer C++ class must reside in a file called *classname*.h, where *classname* is the name of the class. Implementation for all but trivial functions must reside in a file called *classname*.cpp. Note that creating a class using the NetBeans project menu New -> C++ Class... will create empty .h and .cpp files in the proper format.
* All .h files must contain [include guards (Links to an external site.)Links to an external site.](https://en.wikipedia.org/wiki/Include_guard). NetBeans New -> C++ Class... will create these for you.
* In general, functions should be members of classes, not global functions, unless there is a good reason otherwise.
* Each class and each non-trivial function must have a comment block preceding its definition. The comment block for a class should describe its usage. The comment block for a function must describe its usage, all parameters (if any), and the return value (if any). For functions which are class members, the documentation should appear in the .h file, it does not need to be replicated in the .cpp file. Here's an example of a function comment block. The example follows the [Doxygen (Links to an external site.)Links to an external site.](https://en.wikipedia.org/wiki/Doxygen" \t "_blank) standard, but any reasonable comment format is acceptable:

/\*\*

\* ParseCommand - parse a trace file command.

\*

\* @param line return the original command line

\* @param cmd return the command name

\* @param cmdArgs returns a vector of argument bytes

\* @return true if command parsed, false if end of file  
 \* @throws ParseException if invalid command (use what() for error description)

\*/

bool ParseCommand(

std::string &line, std::string &cmd, std::vector<uint32\_t> &cmdArgs);

* All non-trivial class definitions must follow the "[Rule of Five (Links to an external site.)Links to an external site.](http://en.cppreference.com/w/cpp/language/rule_of_three)". This is required for any class which requires any one of the five to operate correctly. This does not apply to classes which will operate correctly with only a constructor defined. If the Rule of Five applies, you must define all Five.
* Functions should be kept small. If a function is over about 100 lines (not including comments), it should probably be refactored into smaller functions.