Good Programming Ideas

Be a Neat Freak

Always pay attention to indentation and spacing in statements. Keep everything clean and re-indent immediately when moving code around. Use a style formatter (e.g., clang-format or your IDE) and let it help you.

Naming is Critical

Picking good names for classes, methods, variables, parameters, files, and everything else that needs a name is critical to having good, readable code. Avoid abbreviations and use proper CamelCase (for classes/types) and camelCase.

Keep Code Nesting Minimal

Avoid deeply nested code like the plague! Use break, continue, and early return to simplify and un-nest code. Split complicated methods/functions. Look for ways to make code more generic and remove special case handling.

Use Documentation-Generating Comments

Format class and method header comments for auto-doc generation (javadoc in Java, doxygen or other tools in other languages). Use inline comments to explain WHY some code is the way it is, do not just repeat what the code says.

Use Linters and Other Tools

Analysis tools exist for every language, so use them. Turn on all possible compiler warnings, and fix the code to not warn. In the C/C++ world, use a memory analysis tool like valgrind to avoid memory leak problems.

Use Assertions and Logging

Assertions are great checks for your assumptions about how your code is executing; however they should be disabled or at least rendered harmless when you ship. Logging is absolutely necessary in any real project; use a good logging library.