

The Ten Rules of a Good Programme

1 Introduction

Since the world of informatics is evolving at the speed of light. There are more programmers all around and few basics might help beginners in programming. C/C++ basics mostly are considered in this document.

2 Golden Rules

What may be the 10 Golden Rules for programming? Here a first attempt, likely, which may need to be adapted.

- 1. Minimum dependencies (as minimum as possible)
- 2. Smaller code as possible and do not produce bloat.
- 3. A code shall be readable and destined to humans.
- 4. The use of { } might have importance for readability
- 5. Portability, portability,... for any platforms (Mac, MS, BSD, Linux,...).
- 6. Assembler (for one arch), C or C++ should be highly considered before starting a project
- 7. ./configure ; make ; make install was, is and will be forever. Make sure that your programme can be easily compiled by a human. A short Makefile is key.
- 8. Man page - might be but not necessary -
- 9. Do never mix programming languages, i.e. Python, Java, Perl,.. it is ugly and non efficient. Avoid System() call.
- 10. The Golden One: Make sure to have a small programme. Several small programs are better than a big one.

3 Description

A code creation shall be made step by step from an easy point of view and improved daily, simplified constantly. The progress of programming shall be a process of natural simplification of the programme. The code shall look smaller and more efficient during programming process. It is better to use your own libraries made from scratch to allow portability. Avoid bloated libraries that will not work after few years due to different versions. Prefer basic X11 to bloated libraries when possible. Smaller code as possible to Avoid System Bloat and Mistakes !!! Minimum dependencies (as minimum as possible) since library versions will change over time and wont be compatible. A created good programme shall be easy to compile after 5-10 years, still! The last law is taking example from MS Windows OS. MS has a big programme running the whole system. This is taking vast amount of memory and make the system slow. Believe me that the smaller your C programme will be the more efficient it will be. In any case, get a quick check at your processes : ps aux, top or htop, mem,... to see how it performs on cpu usage, memory,... Compilation time is also key. A programme shall be compiled in a good realistic time. The use of pointers should be restricted to max. one level. In C, try to stick to good portability by including string, stdio and stdlib at first and then to extend, only if necessary, your included additional libraries. But really, stick to the minimum, it is really important. Maybe in 10-15 years, you have another platform? You might be then happy about good portability. The shorter the code will be the better, but your own libraries from scratch are welcome since they will extend significantly the portability of your programme to another system.

"No function should be longer than what can be printed on a single sheet of paper (in a standard reference format with one line per statement and one line per declaration.) Typically, this means no more than about 60 lines of code per function." (Src: NASA)

All code must be checked regularly and should pass the analyses with zero warnings.

4 Conclusion

10 Golden Rules for programming might help and guide our process of programming. It may also help to have constantly better developments in informatics, considering numerous types of emerging platforms.