

JPL's 10 rules

1. Control flow must be simple. No goto statements or *recursion* allowed.
2. All loops must have a fixed upper-bound on the number of iterations.
3. No dynamic memory allocation (after initialization).
4. No function may be longer than a single sheet of paper.
5. Minimum of two assertions per function.
6. Declarations at the smallest possible level of scope
7. All return values must be checked by the calling function.
8. Preprocessor-use is restricted.
9. No more than one level of dereference is allowed.
10. All code is compiled from day 1 with no warnings.

G Holzman, "The Power of 10: Rules for Developing Safety-Critical Code."
IEEE Computer, June 2006

Suggestions from *Effective Java*

- ▶ 13: Minimize the accessibility of classes and members.
- ▶ 14: In public classes, use accessor methods, not public fields.
- ▶ 15: Minimize mutability.
- ▶ 16: Favor composition over inheritance.
- ▶ 17: Design and document for inheritance or else prohibit it.
- ▶ 18: Prefer interfaces to abstract classes
- ▶ 19: Use interfaces only to define types.
- ▶ 20: Prefer class hierarchies to tagged classes
- ▶ 21: Use function objects to represent strategies
- ▶ 22: Favor static member classes over nonstatic.

Joshua Bloch, *Effective Java*, Addison-Wesley, 2008. Pg 6 7–108