

Design Rules: Unix Philosophy

Rule of Modularity: **Save Time Debugging Complex Code**

Build programs as simple, replaceable parts connected by well-defined interfaces.

Rule of Clarity: **Make Code as Readable as Possible**

Write to communicate with whoever reads the program.

Rule of Composition: **Break Projects into Small Programs**

Write programs that can communicate easily with other programs.

Rule of Separation: **Allow Policy Change**

Separate the program's mechanisms from the program's policies.

Rule of Simplicity: **"Beautiful Complexities" are Bug-Prone.**

Break up program systems into small, straightforward pieces.

Rule of Parsimony: **Avoid Writing Big Programs**

Small programs are easier to write, optimize, maintain, and delete.

Rule of Transparency: **Reduce Debugging Time**

Write so your thought process can lucidly be seen by future developers working on the project. Use input and output formats that make it easy to identify valid input and correct output.

Rule of Robustness: **Build Robust, Reliable Products**

Easy-to-understand is easier to test for unexpected conditions.

Rule of Representation: **Keep Programs Maintainable**

Complex data is easier to understand than complex program logic.

Rule of Least Surprise: **Build Intuitive, Easy to Use Products**

Design programs that build on users' expected knowledge.

Rule of Silence: **Allow Programs to Get Only the Info they Need**

Design programs so that they do not print unnecessary output.

Rule of Repair: **Prevent Bad Output from Becoming Bad Input**

Design to fail in a way that is easy to localize and diagnose.

Rule of Economy: **Reduce Development Costs of Projects**

Value developer time over machine time.

Rule of Generation: **Reduce Human Errors and Save Time**

Write abstract high-level programs that generate code for you.

Rule of Optimization: **Don't Spend Time on Marginal Gains**

Prototype software before polishing it.

Rule of Diversity: **Make Programs Flexible**

Allow your programs to be used in unintended ways.

Rule of Extensibility: **Enhance the Utility of Your Code**

Allow for easy plugins without modification to the program's architecture by other developers.