Avoid multiple exits from loops.
Make sure your code "does nothing" gracefully.
Test programs at their boundary values.
Program defensively.
10.0 times 0.1 is hardly ever 1.0.
Don't compare floating point numbers just for equality.

Make it right before you make it faster.
Keep it right when you make it faster.
Make it clear before you make it faster.
Don't sacrifice clarity for small gains in "efficiency."
Let your compiler do the simple optimizations.
Don't strain to re-use code; reorganize instead.
Make sure special cases are truly special.
Keep it simple to make it faster.
Don't diddle code to make it faster — find a better algorithm.
Instrument your programs. Measure before making "efficiency" changes.

Make sure comments and code agree.
Don't just echo the code with comments — make every comment count.
Don't comment bad code — rewrite it.
Use variable names that mean something.
Use statement labels that mean something.
Format a program to help the reader understand it.
Indent to show the logical structure of a program.
Document your data layouts.
Don't over-comment.