

SUMMARY OF RULES

This summary is designed to give a quick review of the points we covered in the book. Remember as you read the rules that they were presented in connection with one or more examples — go back and reread the pertinent section if a rule doesn't call them to mind.

To paraphrase an observation in *The Elements of Style*, rules of programming style, like those of English, are sometimes broken, even by the best writers. When a rule is broken, however, you will usually find in the program some compensating merit, attained at the cost of the violation. Unless you are certain of doing as well, you will probably do best to follow the rules.

Write clearly — don't be too clever.

Say what you mean, simply and directly.

Use library functions.

Avoid temporary variables.

Write clearly — don't sacrifice clarity for "efficiency."

Let the machine do the dirty work.

Replace repetitive expressions by calls to a common function.

Parenthesize to avoid ambiguity.

Choose variable names that won't be confused.

Avoid the Fortran arithmetic IF.

Avoid unnecessary branches.

Use the good features of a language; avoid the bad ones.

Don't use conditional branches as a substitute for a logical expression.

Use the "telephone test" for readability.

Use DO-END and indenting to delimit groups of statements.

Use IF-ELSE to emphasize that only one of two actions is to be performed.

Use DO and DO-WHILE to emphasize the presence of loops.

Make your programs read from top to bottom.