Writing and Documenting Code

Structured programming

Structured programming is fairly intuitive to anyone with a systems background – like all the best things it is structured common sense!

The Structure Principle: *The static structure (i.e. spread out on the page) of the program should correspond in a simple way to the dynamic structure (i.e. spread out in time) of the corresponding computation. (Dijkstra, 1968)*

Code Elements

All code operations can be broken into three categories. A key objective is to avoid GO TO "spaghetti". GO TO can be eliminated in most languages (except FORTRAN 77 and a few others).

1. Sequence:
A sequential progression through code

2. Repetition:
DO loop
WHILE loop

3. Selection:
IF block

Validation

Four types of bugs:

1. Syntax Errors
Easy found by any complier, usually spelling, dropped commas etc.

2. Link or Build Errors
Always found by complier, usually a spelling or files saved in the wrong place.

3. Run-Time Errors
Program crashes during calculation. Can be a divide by zero, illegal mathematic operation such as log(0) or a bad data entry file.

4. Logic Errors
The killer – the code runs but calculates the wrong thing. Best avoided by having expected values of variables at each point in the operation against which to compare calculated values. Further aids are independently evaluating all subroutines to ensure that they give sensible results over the full range of expected inputs adding checking clauses at entrance to routines to pick up any calls with out of range variables.

Documentation
Code should be internally and externally documented.

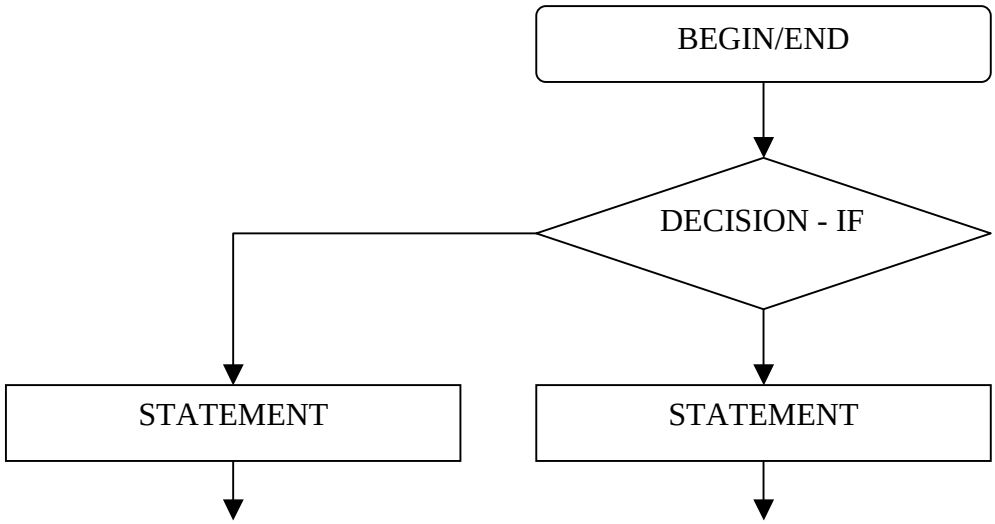
Internal Documentation

1. Title, Author, Date, Contact
2. Variable description INCLUDING UNITS
3. Carefully named variables which are self explanatory
4. Spaces and indentation to illuminate logic
5. Comment lines though out code
6. Description of variables (including units) and function of all modules

External Documentation

1. Structured outputs from program at run time
2. Incorporation of error reports in code
3. Manuals and descriptions (see below)

Generate accurate description of routine. Some of this should be done in design process but it is worth the extra time to tidy up properly for later use – flow chart may or may not be a massive help in development but is very good for quickly re-familiarising.



Use these standard boxes to produce flow charts in Word.
Not as good as purpose written programs but not too tedious and has benefit of portability.

Insert - Picture - Autoshapes - FlowChart

Then from Draw icon at bottom of screen - Grid
And set grid to about 0.5cm with snapping to grid and shapes.
Checking display grid will help with alignment.

Reference:

Numerical Methods for Engineers, Third Edition, S.C. Chapra & R.P. Canale, McGraw-Hill, 1998.

Last Updated: December 3, 2002