

GOOD PROGRAMMING HABITS

1) Program Format

It is a good idea to use indentation when writing programs (SAS or otherwise). While most computer software packages are indifferent to indentation, its use will make your programs *easier for people* to read. The 'block' nature of SAS lends itself to indentation very well. Blank lines between blocks of code also improves readability.

Example 1

BAD:

```
DATA NEW; INFILE 'A:MY.DAT'; INPUT NAME $ AGE CLASS $  
GRADE $; PROC PRINT; VAR NAME GRADE; PROC SORT; BY  
CLASS; PROC PRINT;
```

GOOD:

```
DATA NEW;  
    INFILE 'A:MY.DAT';  
    INPUT NAME $ AGE CLASS $ GRADE $;  
  
PROC PRINT;  
    VAR NAME GRADE;  
  
PROC SORT;  
    BY CLASS;  
  
PROC PRINT;
```

Note that either of these sets of code will produce the same results. The second, however, will be easier to interpret and debug (find & fix errors) if required. One might think that this is inefficient but it is worth it in the long run. In version 8.0 and higher, the

Advanced Editor will automatically indent for you.

2) Comments in Programs

By using special symbols in SAS we can add to a program comment lines which tell anyone reading it what the programmer had in mind when they wrote the code. This can be very useful when programs are passed from person to person or the code is seldom used. The special symbols tell SAS to skip these lines since they are only for us. The format to use these is: `/* any text here */`. Any text written between the `/*` and `*/` will be ignored. Note that this could mean more than one line is passed over.

Example 2

```
/* READ IN DATA FROM EXTERNAL FILE */  
  
DATA NEW;  
  INFILE 'A:MY.DAT';  
  INPUT NAME $ AGE CLASS $ GRADE $;  
  
/* PRINT OUT DATA. (VARIABLES NAME AND  
  GRADE ONLY) .  
*/  
  
PROC PRINT;  
  VAR NAME GRADE;
```

OR

```
DATA NEW;                                /* READ IN */  
  INFILE 'A:MY.DAT';                    /* DATA FROM */  
  INPUT NAME $ AGE CLASS $ GRADE $;     /* FILE */  
  
PROC PRINT;                              /* PRINT NAME */  
  VAR NAME GRADE;                       /* AND GRADE */
```