

# Advanced language features 1

- Macros and tricks for your instrument...
- DECLARE and INITIALIZE

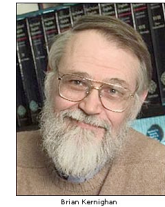


# DECLARE / INITIALIZE

- Use the DECLARE section define user variables and functions.
  - DECLARE %{
    - double myvar;
    - %}
- Use INITIALIZE for initialization of user variables and calculations.
  - INITIALIZE %{
    - myvar = sqrt(PI\*input\_var)\*rand01();
    - %}
- - Both use normal c-syntax.
- BEWARE: (example) What you do in the c-style areas is c-standard, e.g. trigonometric functions from math.h use radians! - McStas placement specifiers work in degrees, etc...



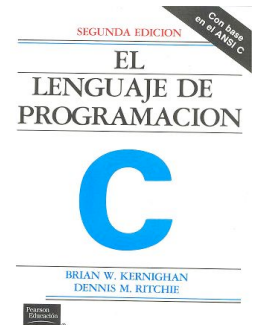
K & R. / GNU



Brian Kernighan



Dennis Ritchie



# %include

- Instrumentfiles can include external c-code or other instrumentfiles... See the examples:
- ILL\_H15\_IN6.instr:%include "monitor\_nd-lib"
- ILL\_H16\_IN5.instr:%include "ILL\_H16.instr"
- ILL\_H25\_IN22.instr:%include "ILL\_H25.instr"
- ILL\_H25\_IN22.instr:%include "templateTAS.instr"
- Used in the DECLARE section

