- For the previous example

1.2345×10^{15} → 1.2345e15 (single, not what we want to do)

→ $\boxed{1.2345d15}$ (correct double precision)

6.432×10^{-8} → 6.432e-8 (single, not correct)

→ $\boxed{6.432d-8}$ (double, correct)

- **IF the constant has digits after the decimal point, you need to use the 'd' descriptor to properly assign a double precision value**

except for powers (sometimes)

Examples

incorrect	correct
a = 6.92	a = 6.92d0
b = 0.0032	b = 0.0032d0 or 3.2d-3

Mixed Mode Calculations

- Assume i, j, k have been declared INTEGER. What is the result of the assignments

i = 1.2
j = -3.8

- since i and j are integers, they can't have decimal parts. The result will be stored as

i = 1
j = -3

(ie, fractional parts are truncated)

- what value would be assigned to k if we do

  i = 1
  j = 2
  k = i / j

$\boxed{k = 0}$ (more exactly, k is an integer 0)

This is an example of <u>integer division</u>. The quotient of 2 integers is also an integer. This is a consequence of the general rule that

if a, b have the same type, then a op b where op = +, −, *, / has the same type as a and b

(Nearly) all programming languages follow this rule.

- If i, j, k, l have been declared to be integers, what is the result of

  i = 1
  j = 2
  K = 2
  l = K * i / j

$\boxed{l = 1}$ K*i = int 2 * int 1 = int 2
K*i/j = int 2 / int 2 = int 1

  i = 1
  j = 2
  K = 2
  l = K * (i / j)

$\boxed{l = 0}$ (i / j) done first because of the ( )
(i / j) = 0
K * (i / j) = 0

- Mixed mode arithmetic refers to how calculactions are performed when expressions involving variables of different types are evaluated

If a and b have different types, the result of

a (op) b

is the higher type of a and b (here, a double is higher than an integer). The lower type is first promoted to the higher type before performing the (op)

Example

Suppose i is an integer and a and b are doubles

a = 1.0d0
i = 2
b = a / i

before computing b, i is promoted to a double 2.

b = double / double → double
= 0.5dø

- when assigning variables with expressions, remember that the expression on the right is evaluated before doing the assignment

Example Suppose i, j are integers and a is double

i = 1
j = 2
a = i / j   a = 0 (double)

First, the integer division i / j is done (result is an integer 0). Then this is promoted to a double 0 when doing the assignment.

- How would you do this "correctly", ie suppose you actually need a to be 1/2 ?

use the DBLE funtion to manually promote i, j or both

$$a = \text{DBLE}(i)/j$$
$$= i/\text{DBLE}(j)$$
$$= \text{DBLE}(i)/\text{DBLE}(j)$$

all of these are equivalent and would give $a = 1/2$ in double precision.

Type conversion functions

INT (a) $\longrightarrow$ converts a to an integer by dropping the fractional part (no rounding is done).

This can overflow if a is large

DBLE (i) $\longrightarrow$ promotes i to a double

<u>Example</u> suppose a, b, c are doubles
i, j, k are integers

What value would c have ?

a = 1.0d0 (note: a = 1 also works because there are
b = 2.0d0 b = 2 no digits after the decimal point

i = 2
j = 5
K = -2

$$c = a * (b/k) \cdot (j/k) \cdot i/j$$

b/K = double/int = -1 (double)
j/K = int/int = 5/-2 = -2.5 = -2 (integer)

c = (double 1) · (double -1) · (integer -2) · (integer 2) /(integer 5)
(double -1) · (integer -2) · (integer 2) /(integer 5)
= (double 2) · (integer 2) /(integer 5) = (double 4) /(integer 5) = 0.8 double