//Source1:

**unit** A

foo **do end**

**end**

// Source2: here we add new routine into A

**extend unit** A

goo **do end**

**end**

//Source3: here we extend inheritance (add new base unit and override routine inherited from B)

**extend unit** A **extend** B

**override** too **do end**

**end**

**unit** B

too **do end**

**end**

/\* Source4: here is the caveat with standalone routines – they in fact extend functionality of the first argument type unit.\*/

standAloneRoutine (arg: A) **do end**

//Source5: this source code will be valid if we assemble a program which contains all 5 sources.

// This is object-driven semantics

a **is** A

a.too (); a.foo (); a.goo (); a.standAloneRoutine ()

// This is module-driven semantics

A.too (); A.foo (); A.goo (); A.standAloneRoutine ()

// Standard way of calling standalone routines

standAloneRoutine (a)