Import JJIO

Class Complex
-- Does return magnitude of a complex number
Function magnitude (none) ofType real is public
  Box result ofType real
  -- Does Complex numbers
  -- Name A Nonono Us
  -- Does return magnitude of a complex number
  -- Set result = Sqrt ( re * re + im * im)
  EndFunction magnitude

Box re ofType real is private -- Real part
Box im ofType real is private -- Imaginary

Constructor Complex (r, i) is public
  Slot r ofType real
  Slot i ofType real
  -- Does construct, create, initialize
  Set re = r
  Set im = i
  EndConstructor Complex

Function plus (that) ofType Complex is public
  Slot that ofType Complex
  Box result ofType Complex
  -- Does add complex numbers
  New result ofType Complex with
    ( re + that.re, im + that.im )
  EndFunction plus

Routine add (x, y) is public
  Slot x ofType Complex
  Slot y ofType Complex
  -- Does add two complex numbers
  Set re = x.re + y.re
  Set im = x.im + y.im
  EndRoutine add

Function times (it) ofType Complex is public
  Slot it ofType Complex
  Box result ofType Complex
  -- Does multiply complex numbers
  Box realPart ofType real
  Box imagPart ofType real
  New result ofType Complex with
    ( realPart = re * it.re - im * it.im
    Set realPart = re * it.re - im * it.im
    Set imagPart = re * it.im + im * it.re
    New result ofType Complex with
      ( realPart, imagPart )
  EndFunction times

Function conjugate (none) ofType Complex is public
  Box result ofType Complex
  -- Does return complex conjugate
  EndFunction conjugate

Routine testComplex (none) is private
  Box A ofType Complex
  Box B ofType Complex
  Box C ofType Complex
  -- Does test the Complex class
  Start
  New A ofType Complex with (1.0, 2.0)
  New B ofType Complex with (3.0, 4.0)
  New C ofType Complex with (0.0, 0.0)
  Set C = A.plus(B)
  Call C.show
  Set C = C.plus(A)
  Call C.show
  Set C = A.times(B)
  Call C.show
  Call C.add with (A, B)
  Call C.show
  Set C = A.times(B)
  Call C.show
  EndRoutine testComplex

Start
  Call C.show
  Call C.add with (A, B)
  Call C.show
  Set C = A.times(B)
  Call C.show