

Fortran 77 : 3. Arithmetic Operators

The mathematical symbols that can be used in Fortran and the corresponding mathematical symbol is given in the following table. The same symbols are used for integer (INTEGER), real (REAL, DOUBLE PRECISION REAL) and complex (COMPLEX, DOUBLE PRECISION COMPLEX) arithmetic. However, for integer arithmetic the result is in integer, so $5/4$ would return '1' (rather than '1.25').

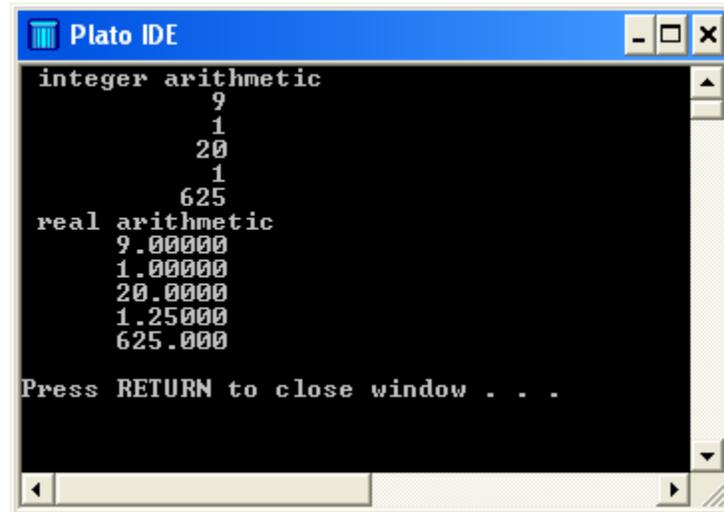
operation	Mathematical symbol(s)	Fortran symbol
addition	$5 + 4$	5+4
subtraction	$5 - 4$	5-4
multiplication	5×4	5*4
division	$5/4$ or $5 \div 4$	5/4
power	5^4	5**4

Integer and real arithmetic

The following program demonstrates integer and real arithmetic in Fortran.

```
C *****  
C ARITHMETIC  
C *****  
      PROGRAM ARITHMETIC  
C integer arithmetic  
      WRITE(*,*) 'integer arithmetic'  
      WRITE(*,*) 5+4  
      WRITE(*,*) 5-4  
      WRITE(*,*) 5*4  
      WRITE(*,*) 5/4  
      WRITE(*,*) 5**4  
C real arithmetic  
      WRITE(*,*) 'real arithmetic'  
      WRITE(*,*) 5.0+4.0  
      WRITE(*,*) 5.0-4.0  
      WRITE(*,*) 5.0*4.0  
      WRITE(*,*) 5.0/4.0  
      WRITE(*,*) 5.0**4.0  
      STOP  
      END
```

Output



```
integer arithmetic
  9
  1
 20
  1
 625
real arithmetic
9.00000
1.00000
20.0000
1.25000
625.000

Press RETURN to close window . . .
```

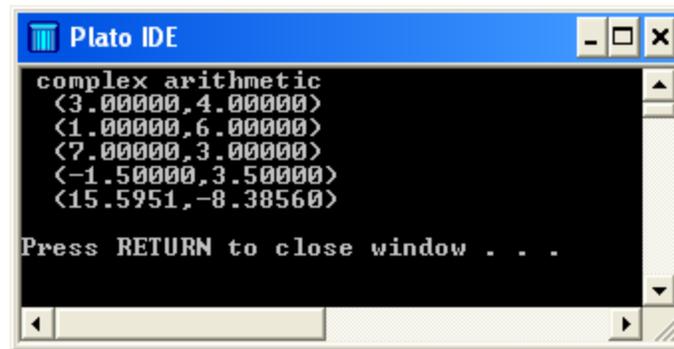
Complex arithmetic

In the following program the operators are applied to complex numbers¹.

```
C *****
C ARITHMETICC
C *****
      PROGRAM ARITHMETIC
C complex arithmetic
      WRITE(*,*) 'complex arithmetic'
      WRITE(*,*) (2.0,5.0)+(1.0,-1.0)
      WRITE(*,*) (2.0,5.0)-(1.0,-1.0)
      WRITE(*,*) (2.0,5.0)*(1.0,-1.0)
      WRITE(*,*) (2.0,5.0)/(1.0,-1.0)
      WRITE(*,*) (2.0,5.0)**(1.0,-1.0)

      STOP
      END
```

¹ [Complex Arithmetic](#)



```
Plato IDE
complex arithmetic
<3.00000,4.00000>
<1.00000,6.00000>
<7.00000,3.00000>
<-1.50000,3.50000>
<15.5951,-8.38560>

Press RETURN to close window . . .
```

Good Practice Tips

1. Choose and use your types carefully. First you need to decide the types that you need to use. For example the heights in metres of a group of people would naturally be best stored as a real number, whereas the number of people in a group would naturally be expressed as an integer. Complex numbers are used when they are required in science and engineering. If you think that a higher precision is necessary then use double precision. Declare all your variables with the appropriate type.
2. Be careful also how you 'mix' types in expressions - for example dividing an integer by a real number. Unless you are 100% sure how the compiler would respond, don't mix your types.