CS 199 Computer Programming

Spring 2018

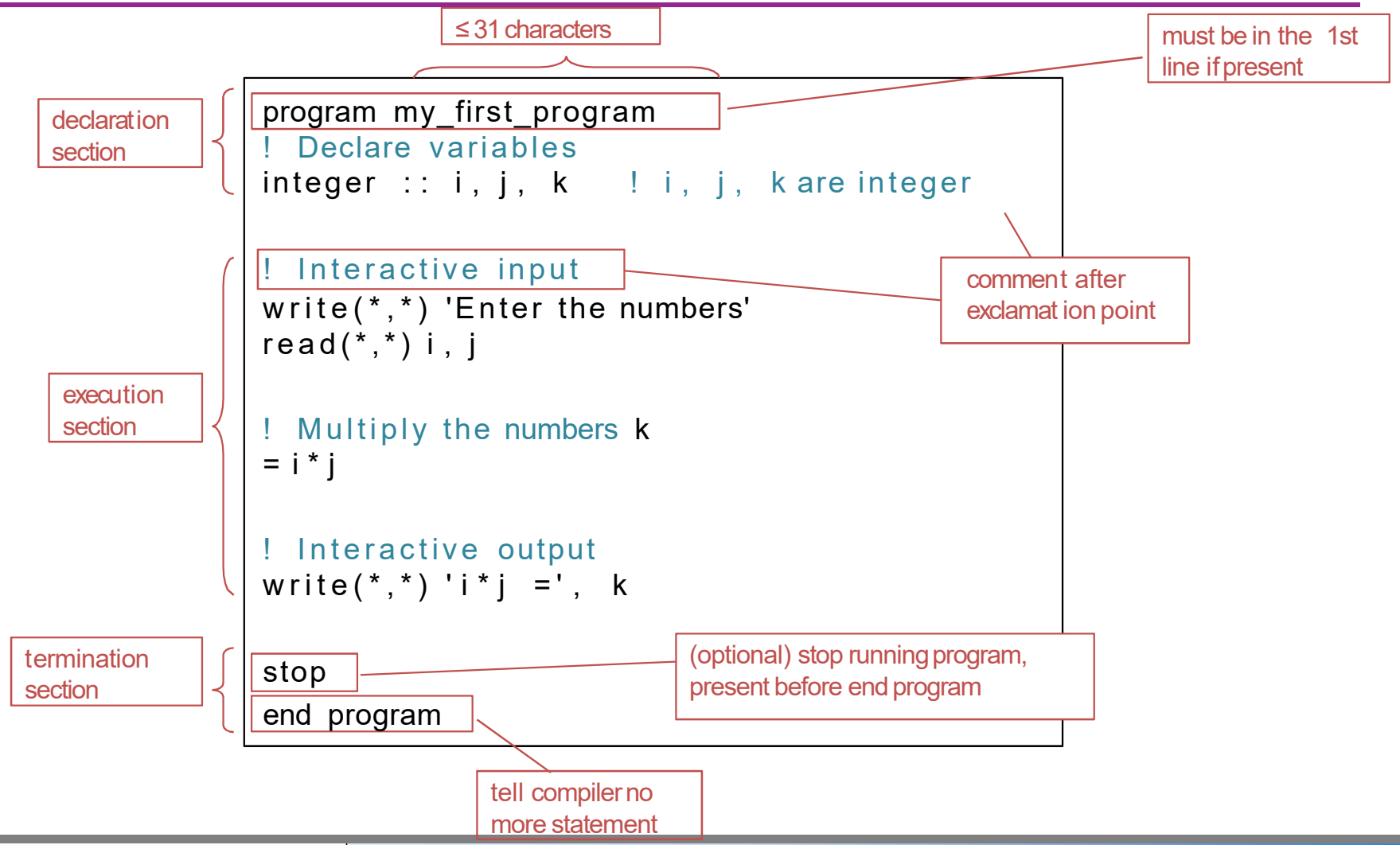
Lecture 4

FORTRAN Operators

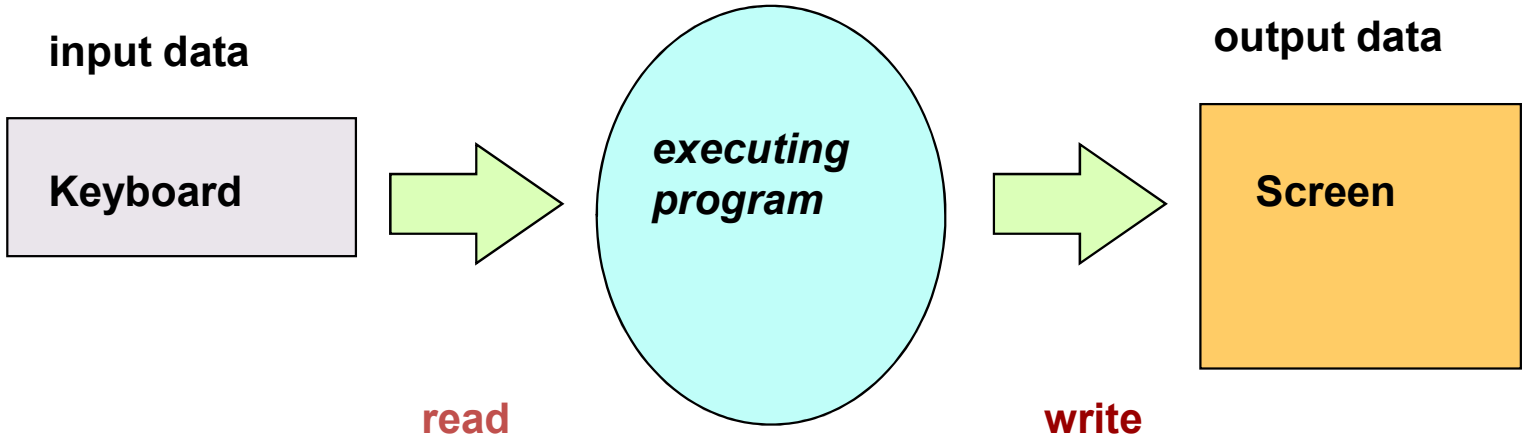
Objectives

- In this chapter, you will learn:
  - To be able to use arithmetic operators.
  - To understand the precedence of arithmetic operators.

Structure of a Fortran Program



Keyboard and Screen I/O



Arithmetic

- Arithmetic is performed with operators.
- Arithmetic operators are listed in following table

FORTRAN operation	Arithmetic operator	Algebraic expression	FORTRAN expression
Addition	+	$f + 7$	`f + 7`
Subtraction	-	$p - c$	`p - c`
Multiplication	*	bm	`b * m`
Division	/	x / y	`x / y`
expontation	**	r^s	`r ** s`

Mathematical formula	FORTRAN Expression
$b^2 - 4ac$	b**2-4*a*c or b*b -4*a*c
x(y+z)	x*(y+z)
$\frac{a+b}{a-c}$	(a+b)/(a-c)
$\frac{1}{\sqrt{x}+x+3}$	1/(x**0.5+x+3)

Results of Arithmetic operators

- Arithmetic operators can be used with any numeric type.
- An **operand** is a number or variable used by the operator e.g.
  - integer1 + integer2
    - + is operator
    - integer1 and integer2 are operands
- The result of an expression is always of the higher type, for example:
  - INTEGER * REAL gives REAL , (3*2.0 is 6.0)
  - REAL * INTEGER gives REAL , (3.0*2 is 6.0)
  - COMPLEX * < anytype > gives COMPLEX

Mixed Type Assignment

- Problems can occur with mixed-type arithmetic. The rules for type conversion are given below:
  - INTEGER = REAL
    - ✓The RHS is evaluated, truncated (all the decimal places removed) then assigned to the LHS.
  - REAL = INTEGER
    - ✓The RHS is evaluated, promoted to be REAL (approximately) and then assigned to the LHS.
- For example:

```
REAL :: a = 1.1, b = 0.1
INTEGER :: i, j, k
i = 3.9          ! i will be 3
j = -0.9         ! j will be 0
k = a – b        ! k will be 1
```

Integer Division

- Division of two integers produces an integer result by truncation (towards zero). Consider:
  - REAL :: a, b, c, d, e
  - a = 1999/1000 **! LHS a is (about) 1.000**
  - b = -1999/1000 **! LHS b is (about) -1.000**
  - c = (1999+1)/1000 **! LHS c is (about) 2.000**
  - d = 1999.0/1000 **! LHS d is (about) 1.999**
  - e =1999/1000.0 **! LHS e is (about) 1.999**

Rules of operator precedence

Some arithmetic operators act before others (e.g., multiplication before addition)

Opertors	Opeertions	Order of evaluation (precedence)
()	Parentheses	Evaluated first. If the parentheses are nested, the expression in the innermost pair is evaluated first. If there are several pairs of parentheses "on the same level" (i.e., not nested), they are evaluated left to right.
******	Expontation	Evaluated second.
*** or /**	Multiplication Division	Evaluated third. If there are several, they are evaluated left to right.
+ or **-**	Addition Subtraction	Evaluated last. If there are several, they are evaluated left to right.

Operator Precedence

An example to understand operator precedence.

```
20 -  4 / 5  * 2   +   3 * 5  / 2
```

```
              (4 / 5)    0.8 → 0
             ((4 / 5) * 2)    0*2→0
             ((4 / 5) * 2)      (3 * 5)              15
             ((4 / 5) * 2)     ((3 * 5) / 4)      3.75→ 3
       (20 -((4 / 5) * 2))     ((3 * 5) / 4)      20-0→20
       (20 -((4 / 5) * 2)) +   ((3 * 5) / 4)      20+3→23
```

Math functions

- Allow the programmer to perform common mathematical calculations
- **Example**

  Write(*,*) sqrt(900.0);

  - Calls the **sqrt** (square root) function.
  - The preceding statement would print **30**
  - The **sqrt** function takes an argument of type **real**
  - And returns a result of type **real**
- **Function call arguments can be**
  - Constants **sqrt(4.0)**
  - Variables **sqrt(x)**
  - Expressions **sqrt(sqrt(x)) , sqrt(3y + 6)**

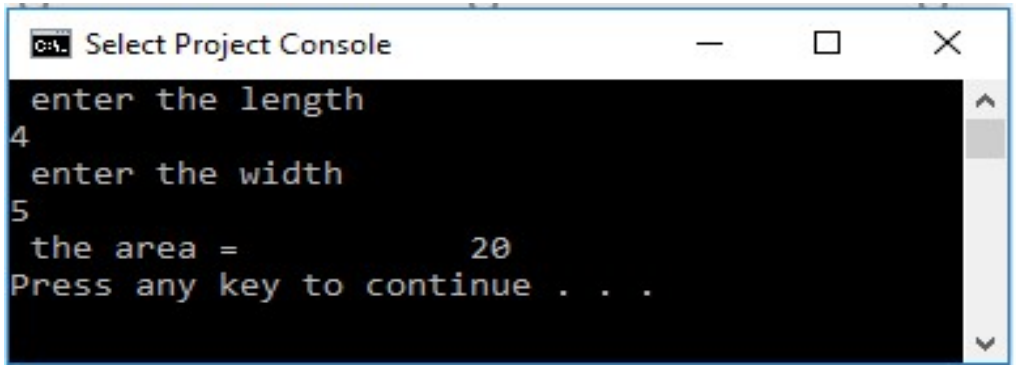
Math functions

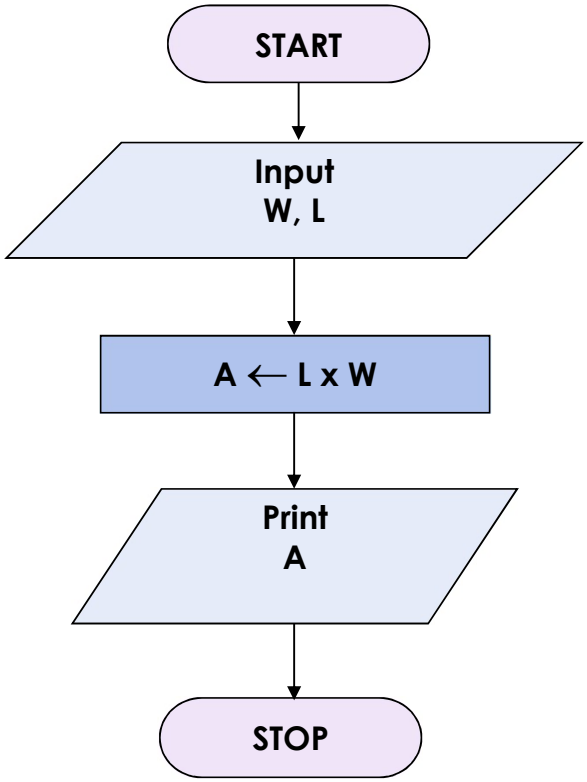
function	description	example
Cos(x)	cosine where x is in radians	Cos(0.0)=1.000
Sin(x)	sine where x is in radians	sin(0.0)=0.000
Tan(x)	tangent where x is in radian	tan(0.0)=0.000
Log(x)	natural logarithm of x	Log(100.0)=4.60517025
Log10(x)	logarithm base 10 of x	Log10(100.0)=2.00
Sqrt(x)	the square root of x	Sqrt(9.0)=3.000
Abs(x)	absolute value	Abs(-9)=9
MOD(a,p)	remainder function	Mod(14,4)=2
CEILING(a)	smallest INTEGER greater than or equal to REAL number	Ceiling(2.3)=3
FLOOR(a)	biggest INTEGER less than or equal to REAL number	Floor(2.3)=2

Example1

Write program that will read the two sides of a rectangle and calculate its area.

```
integer :: l,w,a
write(*,*) 'enter the length'
read (*,*) l
write(*,*) 'enter the width'
read (*,*) w
a=l*w
write (*,*)'the area = ',a
end
```





Example 2

Write an algorithm and draw a flowchart that will calculate the roots of a quadratic equation $ax^2 + bx + c = 0$

```
integer :: a,b,c
real :: d,x1,x2
read (*,*) a,b,c
d= sqrt(b**2-4.0*a*c)
x1= (-b+d)/(2*a)
x2= (-b-d)/(2*a)
write (*,*)'the first root ‘ ,x1
write (*,*) 'the second root ‘ ,x2
end
```

```
Project Console
1
-5
6
 the first root     3.00000000
 the second root    2.00000000
Press any key to continue . . .
```