

# Computers in Engineering COMP 208

Subroutines Michael A. Hawker

### Subprograms

- Functions are one type of subprogram in FORTRAN
- Another type of subprogram FORTRAN allows is called a subroutine
- There are many similarities between them and we must be careful not to confuse the two types of subprograms

Subroutines

## Subroutines

•Oct. 11th, 2007

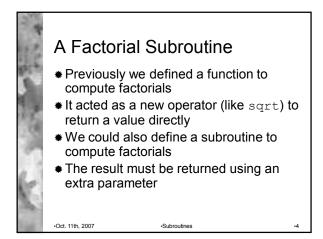
- \* Subroutines are used to define new actions
- Unlike functions, they do not return values
- They can modify the values of arguments or return values indirectly through the arguments

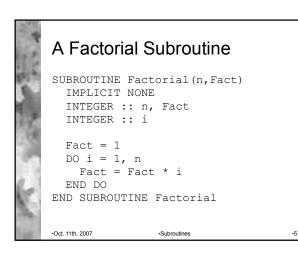
Subroutines

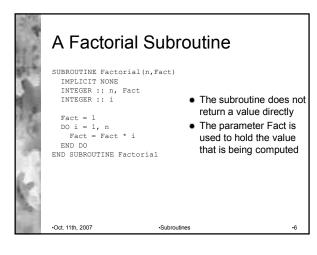
 For example a Sort subroutine may take an array as an argument and return the array with the values in sorted order

•Oct. 11th, 2007

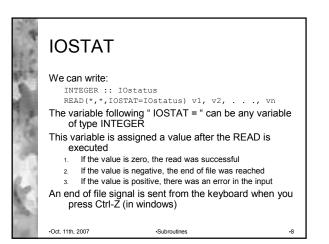
•2



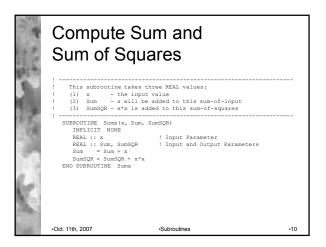


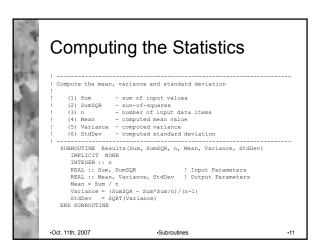


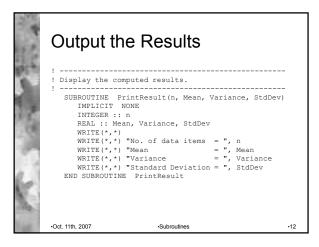
三十二	Computing Statistics
1	! Read an indeterminate number of real values and compute their mean, ! variance and standard deviation.
-	PROGRAM MeanVariance IMPLICIT NONE INFEGER :: Number IOstatus REAL :: Data, Sum, Sum2 REAL :: Mean, Var, Std
	Number = 0 Sum = 0.0 Sum2 = 0.0
6	READ(*,*,IOSTAT-IOstatus) Data IF (IOstatus < 0) EXIT Number = Number + 1 CALL Sums(Data, Sum, Sum2) END DO
-la	CALL Results(Sum, Sum2, Number, Mean, Var, Std) CALL PrintResult(Number, Mean, Var, Std) END PROGRAM MeanVariance
1 pet	•Oct. 11th, 2007 •Subroutines •7



うちの	Computing Statistics	
1	! Read an indeterminate number of real values and compute their mean, ! variance and standard deviation.	
1.0	PROGRAM MeanVariance IMPLICIT NONE INTEGER :: Number, IOstatus REAL :: Data, Sum, Sum2 REAL :: Mean, Var, Std Number = 0	
	Sum = 0.0 Sum2 = 0.0 DO	
K	READ(*,*,1037AT=TOstatus) Data IF (Totatus < 0) EXIT Number = Number + 1 CALL Sums(Data, Sum, Sum2) END DO CALL Results(Sum, Sum2, Number, Mean, Var, Std) CALL PreintKesult(Number, Mean, Var, Std)	
N Los	END PROGRAM MeanVariance +Oct. 11th, 2007 +Subroutines +9	)







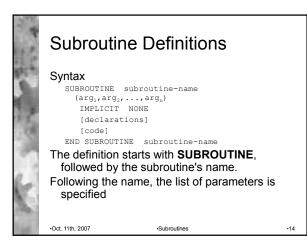


### Advantages of Subprograms Why use subprograms? • Supports top down program design to simplify developing complex programs • Allows for independent testing of subtasks • Allows us to develop reusable code • Isolates the program from side effects that may be caused by the subprogram

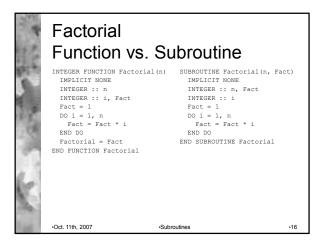
Subroutines

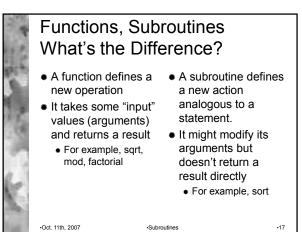
•13

•Oct. 11th, 2007



# Function or Subroutine? Factorial takes a single argument and returns a single value Defining it as a function seems more natural Defining it as a subroutine is more forced Sometimes we don't have a choice





# Functions, Subroutines What's the Difference?

- A function must assign a value to the dummy variable which is the name of the function
- The name of the subroutine is not a dummy variable and is not assigned a value

•Oct. 11th, 2007

Subroutines

### Functions, Subroutines What's the Difference?

- A function is invoked implicitly by using it in an expression.
- After executing, it returns a value to be used in evaluating the expression
- A subroutine is called explicitly. It appears in the program where a statement can appear
- After executing it just returns
- The argument values may have changed

•19

•20

•Oct. 11th, 2007

•Oct. 11th, 2007

### Definition vs. Usage

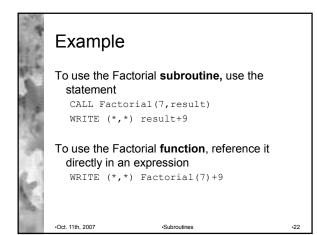
• We have discussed how to define subprograms and the difference between function and subroutine definition

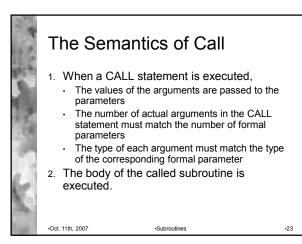
Subroutines

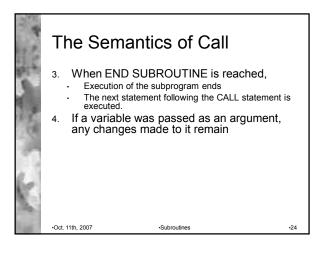
- \* Definitions are a one time thing
- Once defined, the subprograms can be used throughout the program
- The way functions and subroutines are used differs

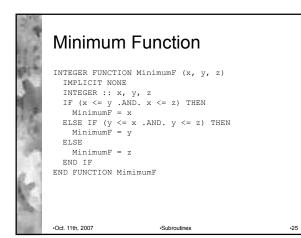
Subroutines

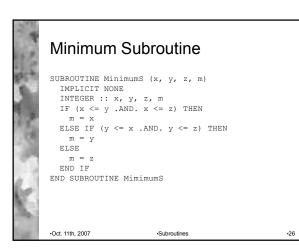
### 





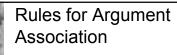






Examp	les of Use
INTEGER :: READ (*,*)	a, b, c, result a, b, c
	mS(a,b,c,result) "The minimum of ", a, b, c, " is: ",& result
WRITE (*,*)	"The minimum of ", a, b, c, " is: ",& MinimumF(a,b,c)
•Oct. 11th, 2007	•Subroutines •27





 The rules for associating arguments with formal parameters are the same as the rules we described for functions

Subroutines

•28

•29

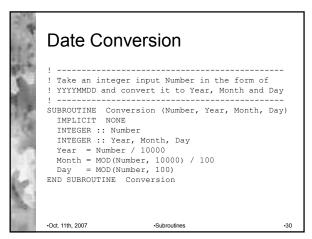
### **Multiple Results**

•Oct. 11th, 2007

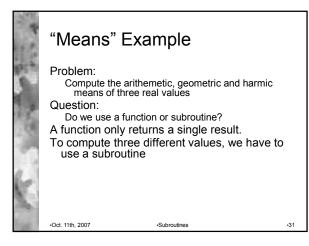
•Oct. 11th, 2007

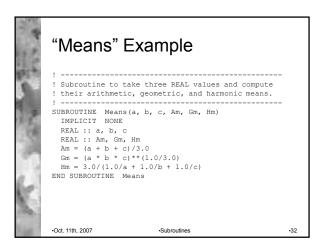
- Sometimes we want operators that return multiple results
- FORTRAN functions cannot be used because they can only return a single value
- In such cases (as in the next example) we must use subroutines

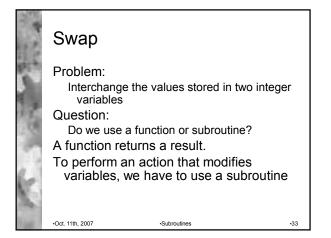
Subroutines

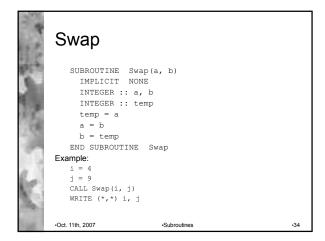














### Array Parameters

•Oct. 11th, 2007

- When we declare an array in a program, we must specify its size
- The compiler allocates storage for the specified number of memory cells
- When we write a subprogram definition to process an array, we want it to be generic
- That is, we want to be able to use it with different arrays, possibly of different sizes

Subroutines

•35

Minimum Value in an Array REAL FUNCTION Min (A, n) IMPLICIT NONE INTEGER :: n INTEGER :: 1 REAL :: A(n) Min = A(1) DO I = 2, n IF (A(I) < Min) Min = A(I) END DO RETURN Min END FUNCTION Min

