Functions in Fortran

Victor Eijkhout and Charlie Dey

spring 2017

Subprogram basics

Subprograms in contains clause

l:contains

```
Program foo
  < declarations>
  < executable statements >
  Contains
    < subprogram definitions >
End Program foo
```

Subroutines

l:subroutine

```
subroutine foo()
  implicit none
  print *,"foo"
  if (something) return
  print *,"bar"
end subroutine foo
```

- Looks much like a main program
- Ends at the end, or when `return` is reached
- Activated with

  ```
  call foo()
  ```

Functions

l:ffunction

```
logical function test(x)
  implicit none
  real :: x

  test = some_test_on(x)
  return ! optional, see above
end function test
```

- No `call`, instead

  ```
  y = test(x)
  ```

- Assign result to function name

Subprogram arguments

Arguments are defined in subprogram body:

```
subroutine f(x,y,i)
  implicit none
  integer,intent(in) :: i
  real(4),intent(out) :: x
  real(8),intent(inout) :: y
  x = 5; y = y+6
end subroutine f
! and in the main program
call f(x,y,5)
```

Modules

Module definition

```
Module FunctionsAndValues
  implicit none

  real(8),parameter :: pi = 3.14

contains
  subroutine SayHi()
    print *,"Hi!"
  end subroutine SayHi

End Module FunctionsAndValues
```

Module use

```
Program ModProgram
  use FunctionsAndValues
  implicit none

  print *,"Pi is:",pi
  call SayHi()

End Program ModProgram
```