Fortran Study

Carlos M.C.G. Fernandes

July 3, 2022

Abstract

1 Variables

```
program variables
    ! this statement tells the compiler
    ! that all variables will be
    ! explicitly declared
    implicit none

    integer :: amount
    real :: pi
    complex :: frequency
    character :: initial
    logical :: isOkay

    amount = 10
    pi = 3.1415927
    frequency = (1.0, -0.5)
    initial = 'A'
    isOkay = .false.

    print *, 'The value of amount (integer) is: ', amount
    print *, 'The value of pi (real) is: ', pi
    print *, 'The value of frequency (complex) is: ', frequency
    print *, 'The value of initial (character) is: ', initial
    print *, 'The value of isOkay (logical) is: ', isOkay

end program variables
```

1.1 Read values

```
program read_value
    implicit none

    integer :: age

    print *, 'Please enter your age: '
    read(*,*) age

    print *, 'Your age is: ', age

end program read_value
```

1.2 Float precision

```
program float
    use, intrinsic :: iso_fortran_env, only: sp=>real32, dp=>real64
    implicit none

    real(sp) :: float32
    real(dp) :: float64

    float32 = 1.0_sp ! Explicit suffic for literal constants
    float64 = 1.0_dp

end program float
```

1.3 Compute cylinder volume

```
program arithmetic
    implicit none

    real :: pi
    real :: radius
    real :: height
    real :: area
    real :: volume

    pi = 3.1415927

    print *, 'Enter cylinder base radius: '
    read(*,*) radius
```

```
    print *, 'Enter cylinder height: '
    read(*,*) height

    area = pi * radius**2.0
    volume = area * height

    print *, 'Cylinder radius is: ', radius
    print *, 'Cylinder height is: ', height
    print *, 'Cylinder base area is: ', area
    print *, 'Cylinder volume is: ', volume

end program arithmetic
```

2 Arrays

```
program arrays
    implicit none

    ! 1D integer array
    integer, dimension(10) :: array1

    ! An equivalent array declaration
    integer :: array2(10)

    ! 2D real array
    real, dimension(10, 10) :: array3

    ! Custom lower and upper index bounds
    real :: array4(0:9)
    real :: array5(-5:5)

end program arrays
```

2.1 Slicing

```
    program array_slice
    implicit none

    integer :: i
    integer :: array1(10) ! 1D integer array of 10 elements
    integer :: array2(10, 10) ! 2D integer array of 100 elements
    character :: exiter
```

```
    ! Array constructor
    array1 = [1, 2, 3, 4, 5, 6, 7, 8, 9, 10]
    print *, 'Array 1: '
    print *, array1

    ! Implied loop constructor
    array1 = [(i, i = 1, 10)]
    print *, 'Implied loop constructor: '
    print *, array1

    ! Set all elements to zero
    array1(:) = 0
    print *, 'Set all elements to zero: '
    print *, array1

    ! Set first five elements to one
    array1(1:5) = 1
    print *, 'Set first five elements to one:'
    print *, array1

    ! Set all elements first five to one
    array1(6:) = 1
    print *, 'Set all elements first five to one:'
    print *, array1

    ! Print out elements at odd indices
    print *, 'Print out elements at odd indices:'
    print *, array1(1:10:2)

    ! Print out the first column in a 2D array
    Print *, 'Print out the first column in a 2D array:'
    print *, array2(:,1)

    ! Print an array in reverse
    Print *, 'Print an array in reverse:'
    print *, array1(10:1:-1)

    ! Print array 2
    print *, 'Array 2:'
    print *, array2

    print *, 'Press any letter+Enter to exit:'
    read(*,*) exiter

end program array_slice
```