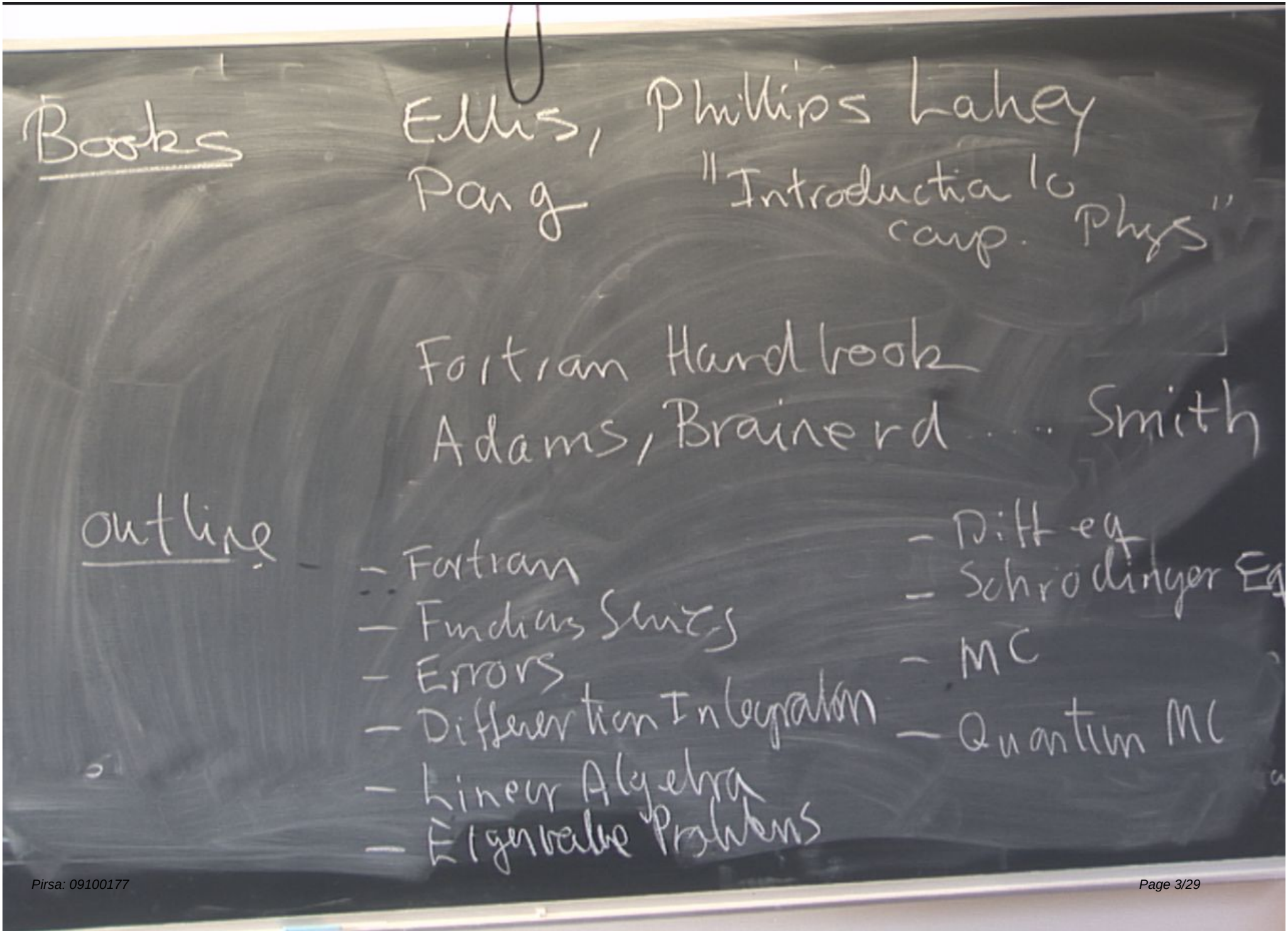
Title: Scientific Computation (PHYS 608) - Lecture 1

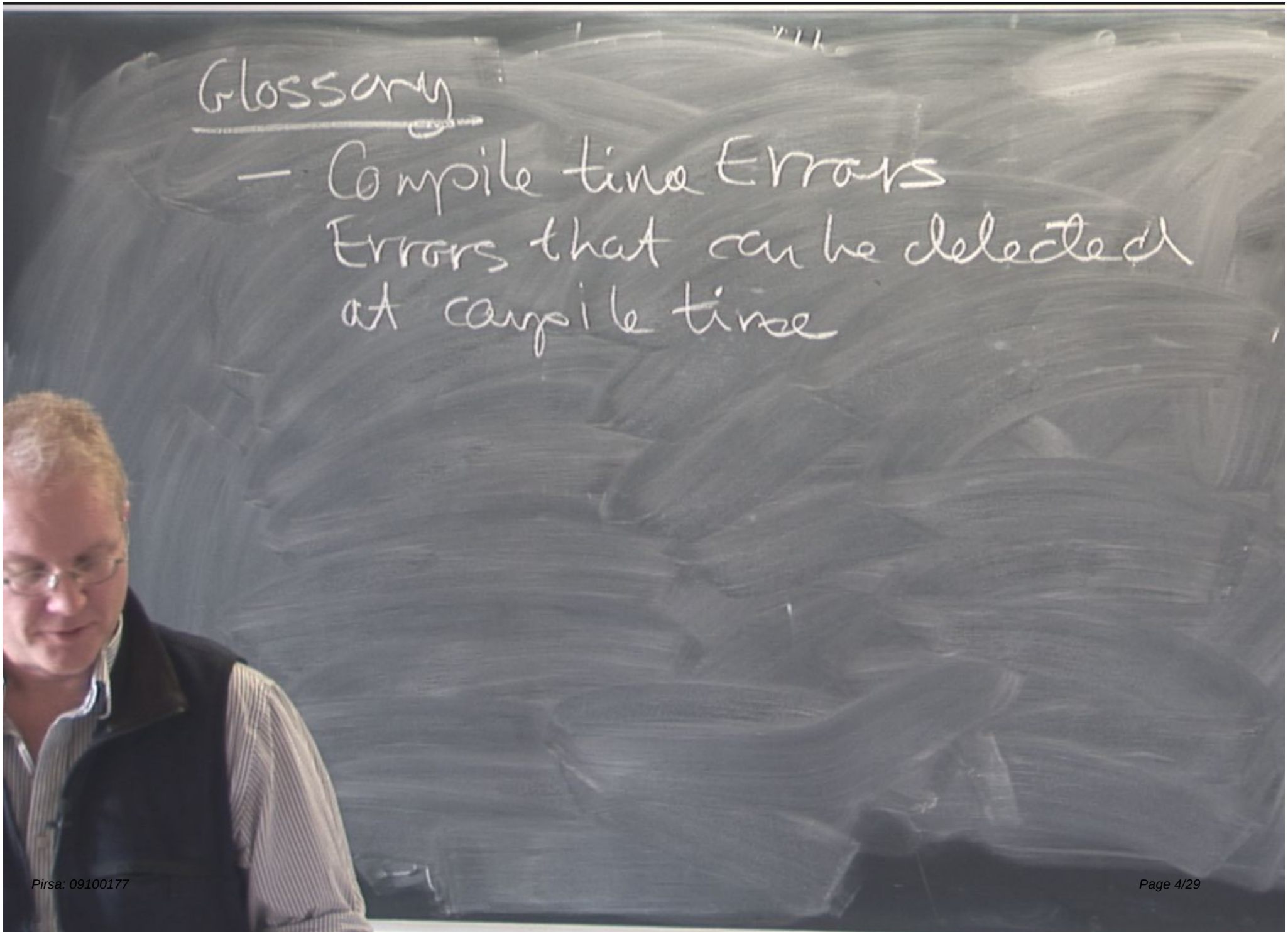
Date: Oct 26, 2009 10:30 AM

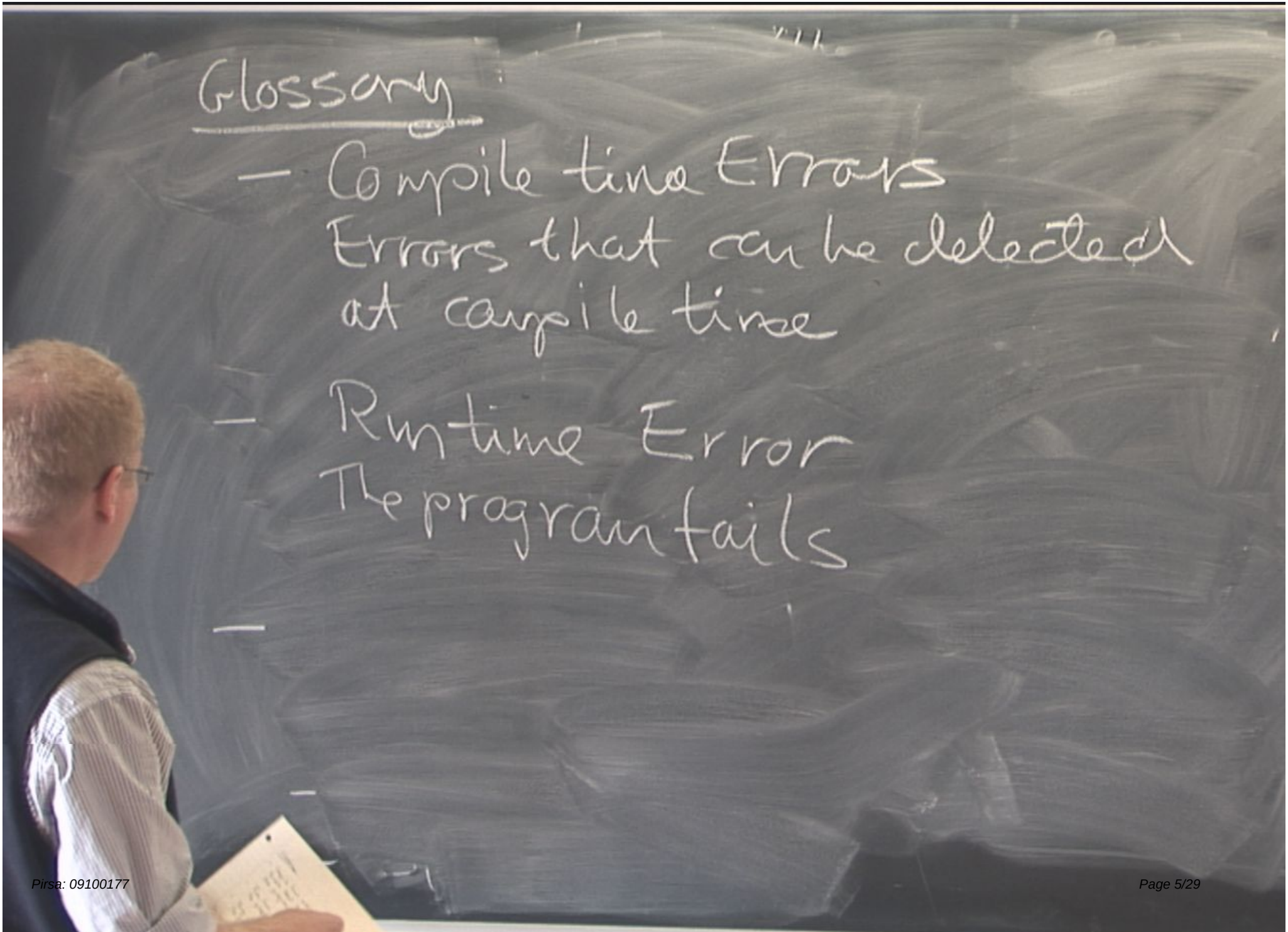
URL: http://pirsa.org/09100177

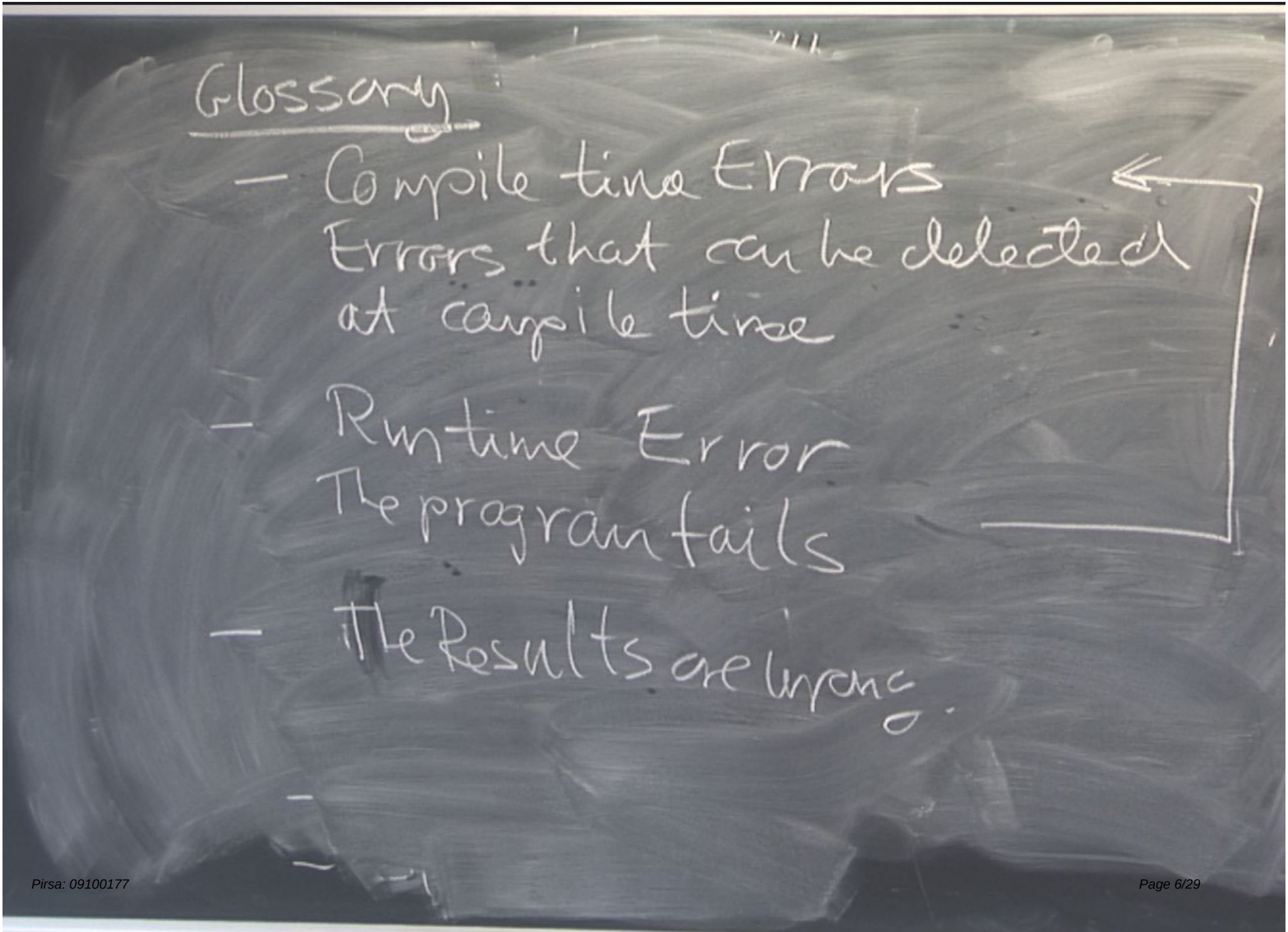
Abstract:

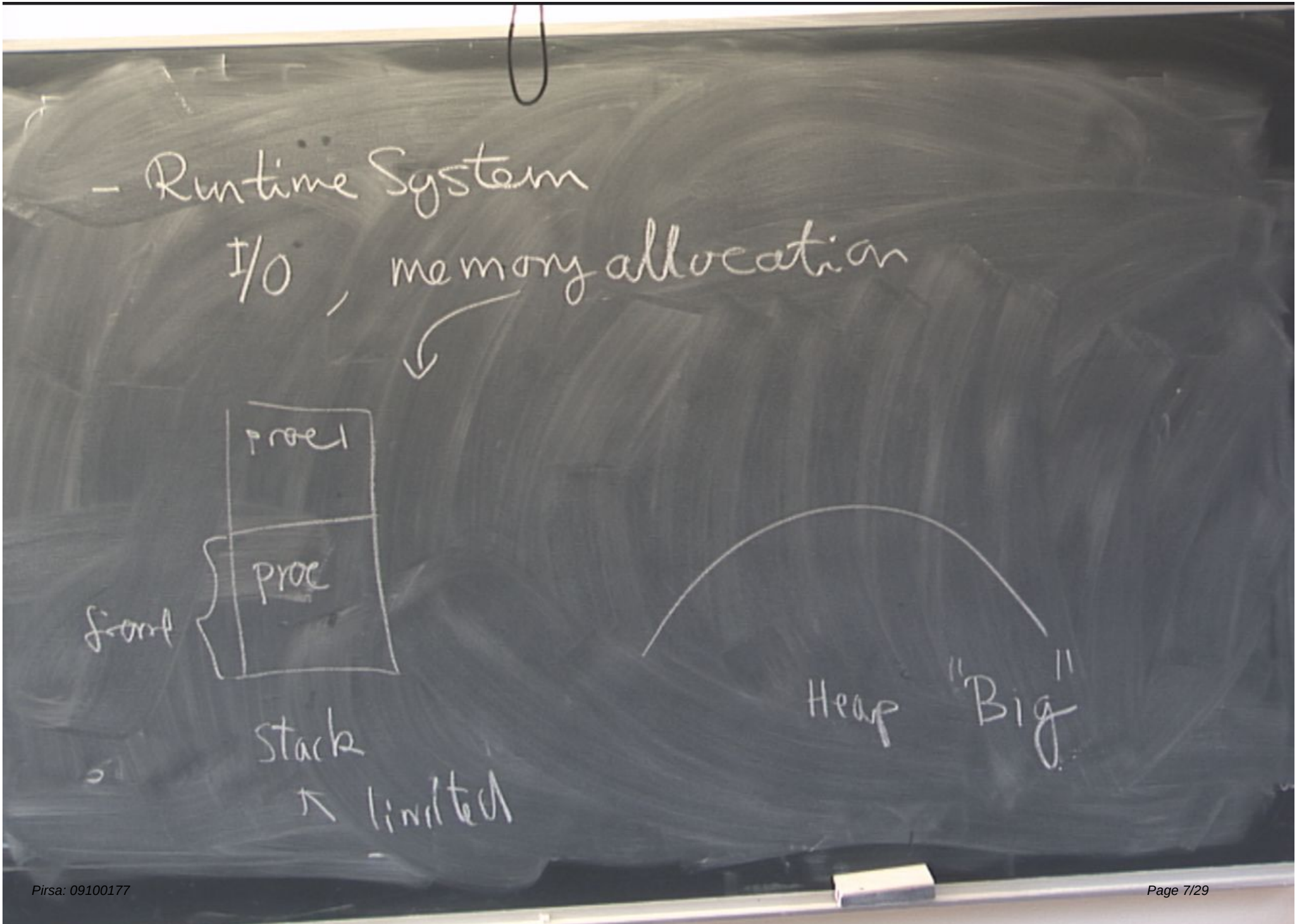

PERIMETER
INSTITUTE FOR THEORETICAL PHYSICS

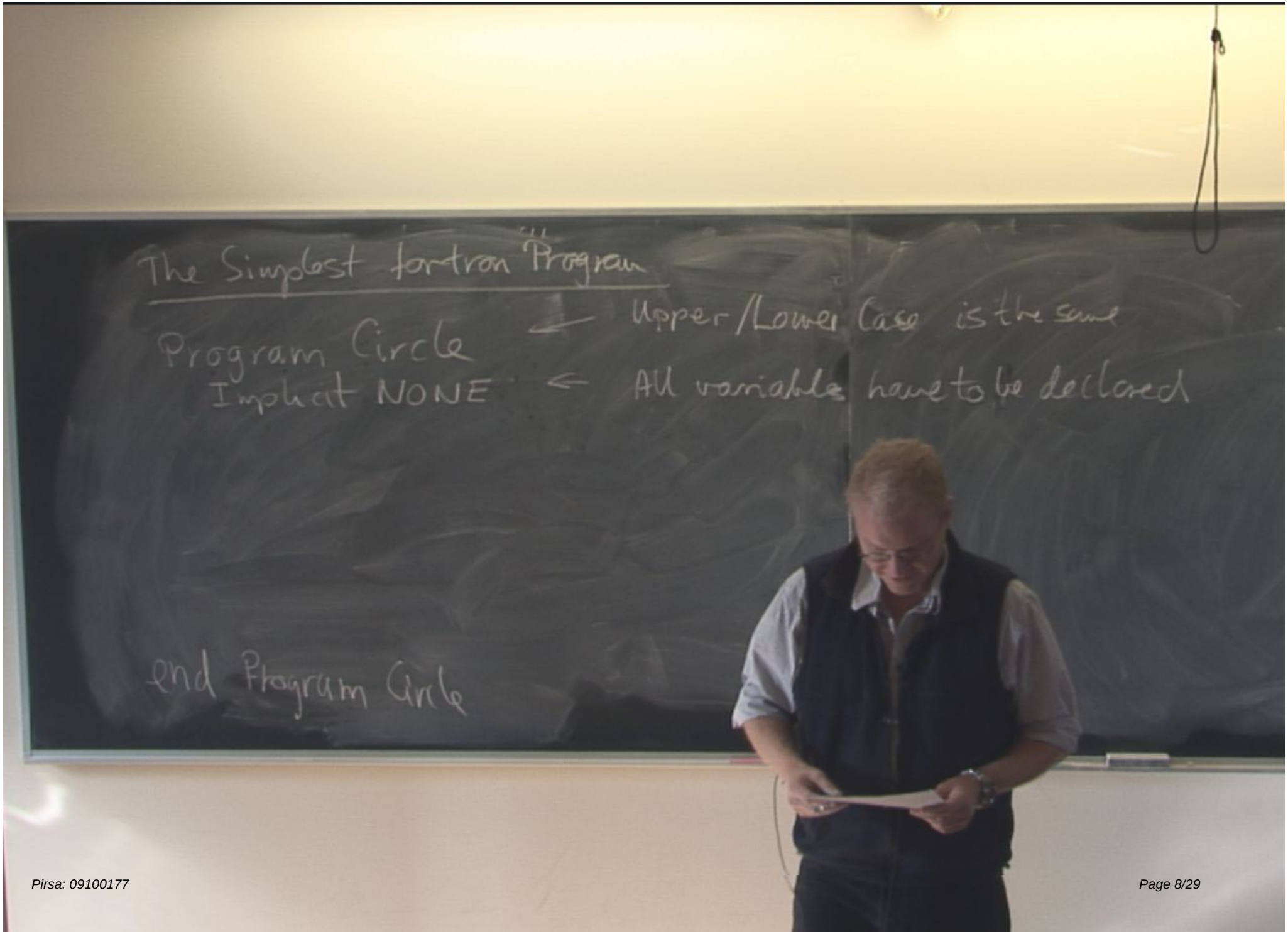

Books
Ellis, Phillips Lahey
Pang "Introductia to comp. Phys"
Fortran Handbook
Adams, Brainerd ... Smith
outline
- Fortran
- Functions Series
- Errors
- Differentiation Integration
- Linear Algebra
- Eigenvalue Problems
- Diff eq
- Schrodinger Eq
- MC
- Quantum MC

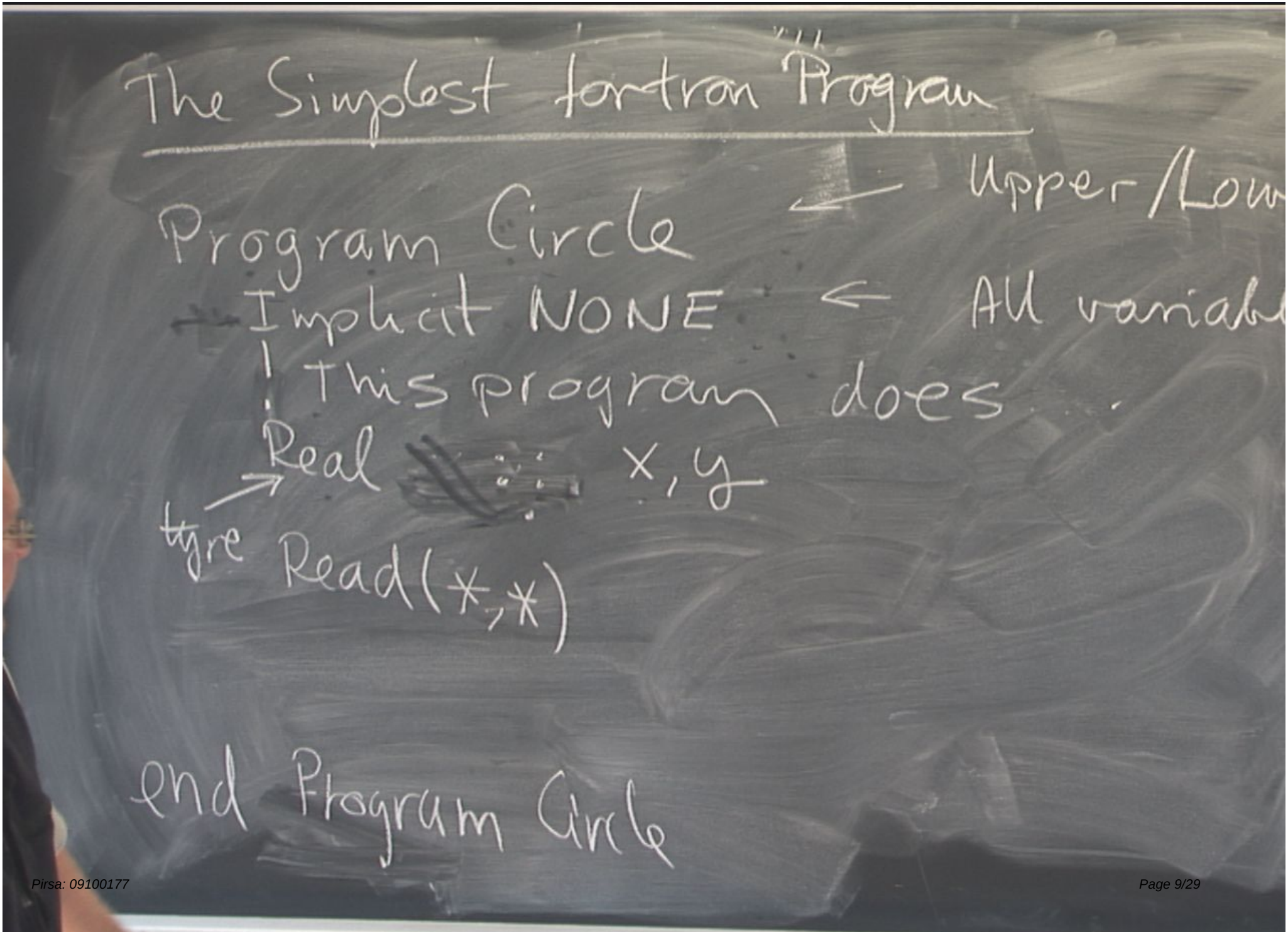

Glossary
– Compile time Errors
Errors that can be detected
at compile time


Glossary
- Compile time Errors
Errors that can be detected at compile time
- Runtime Error
The program fails
-
-


Glossary
- Compile time Errors
Errors that can be detected at compile time
- Runtime Error
The program fails
- The Results are wrong.


- Runtime System
I/O , memory allocation
proc1
proc
frame
Stack
limited
Heap "Big"


The Simplest fortran "Program"
Program Circle
Implicit NONE
Upper/Lower Case is the same
All variables have to be declared
end Program Circle


The Simplest fortran "Program"
Program Circle
Upper/Low
Implicit NONE
All variabl
This program does ...
Real :: x, y
type
Read(*,*)
end Program Circle

The Simplest fortran "Program"

```
Program Circle          <- Upper/Low
  Implicit NONE         <- All variabl
  ! This program does ...
  Real :: x, y
  Read(*,*) x
  y = sqrt(1 - x*x)
  write(*,*) y
end Program Circle
```

type

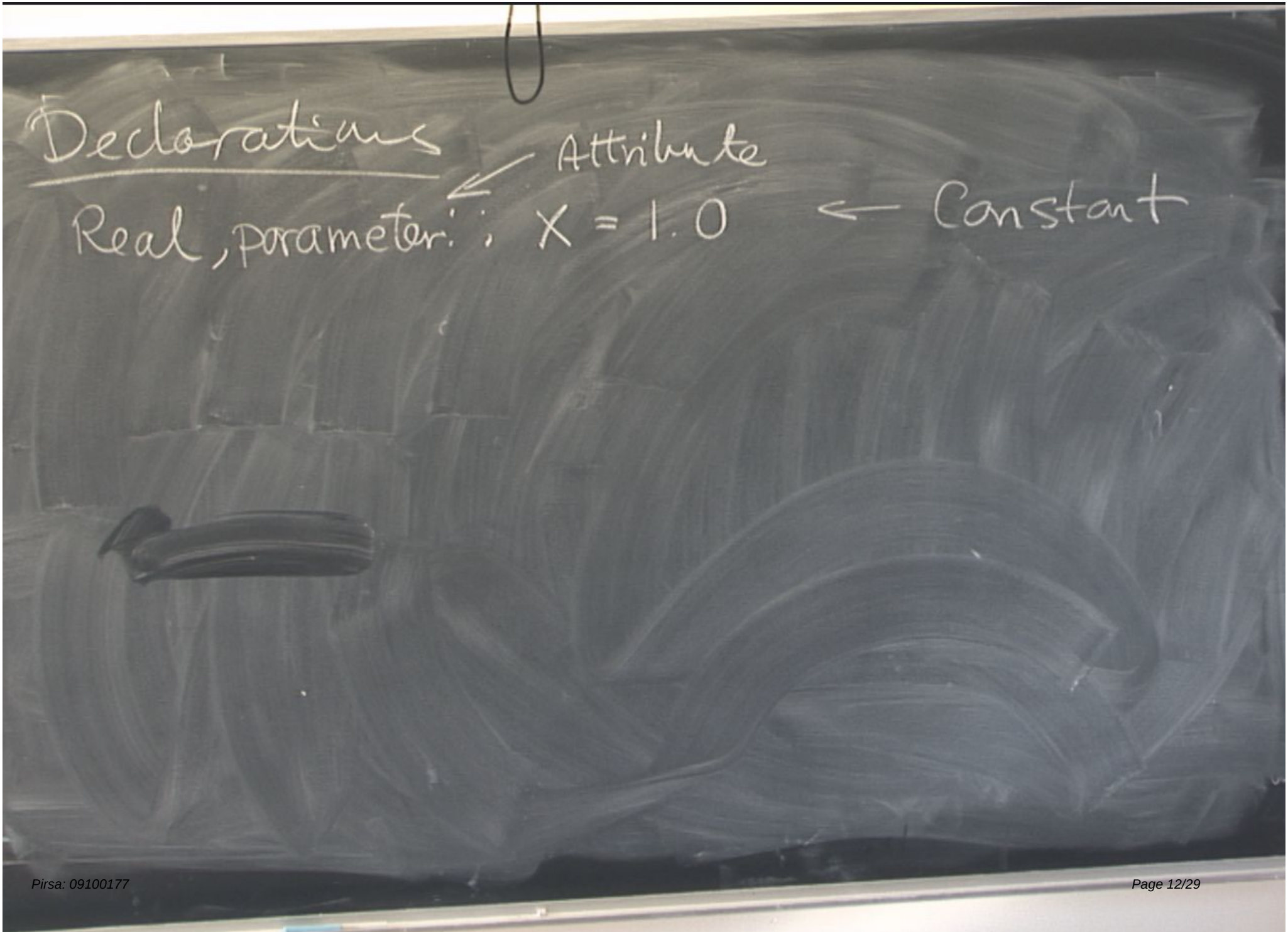
The Simplest fortran Program

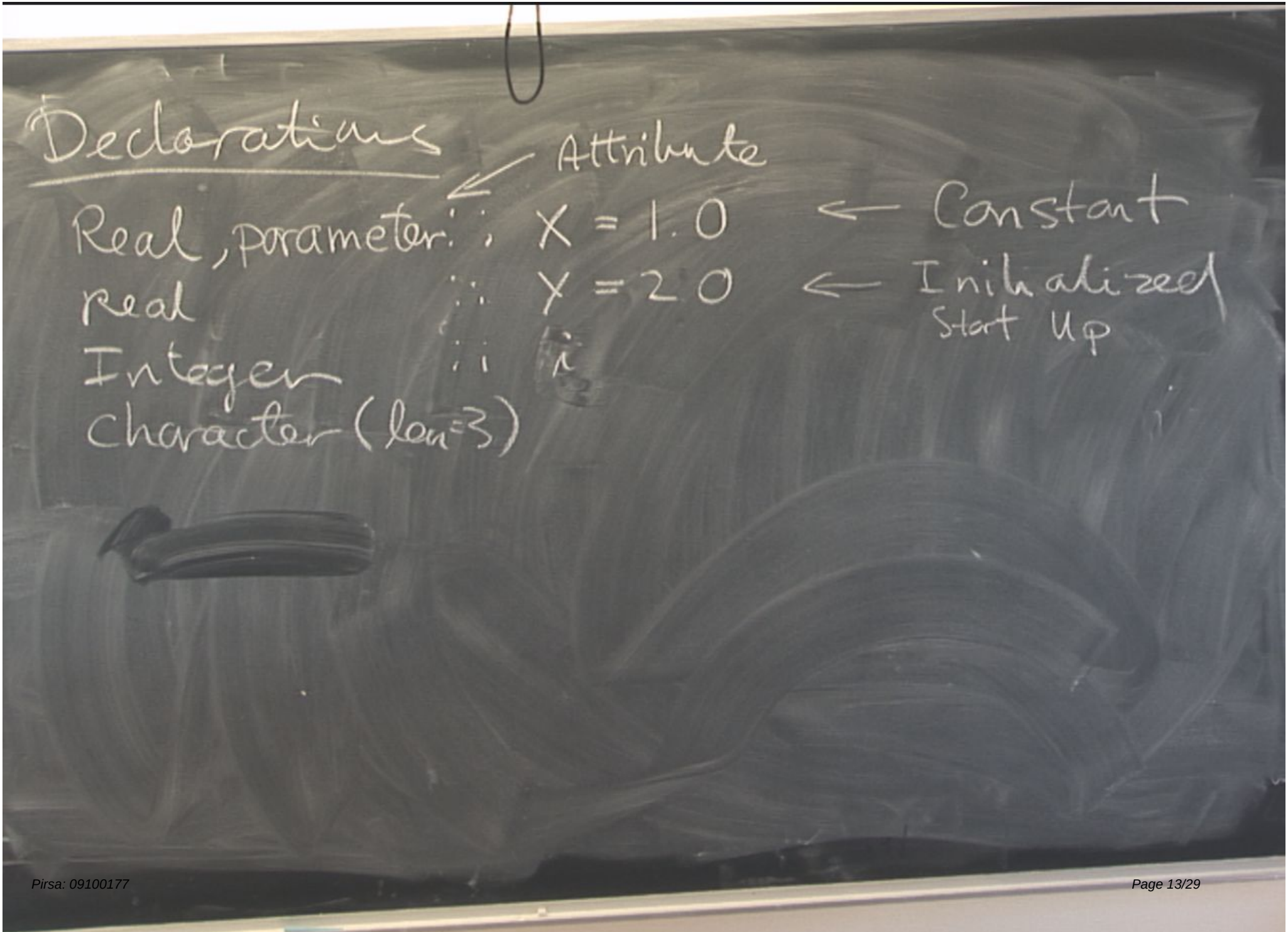
```
Program Circle          <- Upper/Low
  Implicit NONE         <- All variabl
  ! This program does ...
  Real :: x, y
  Read(*,*) x
  y = sqrt(1 - x*x)
  write(*,*) y
end Program Circle
```

declaration part

tyre

Statements


Declarations
Attribute
Real, parameter :: X = 1.0
Constant

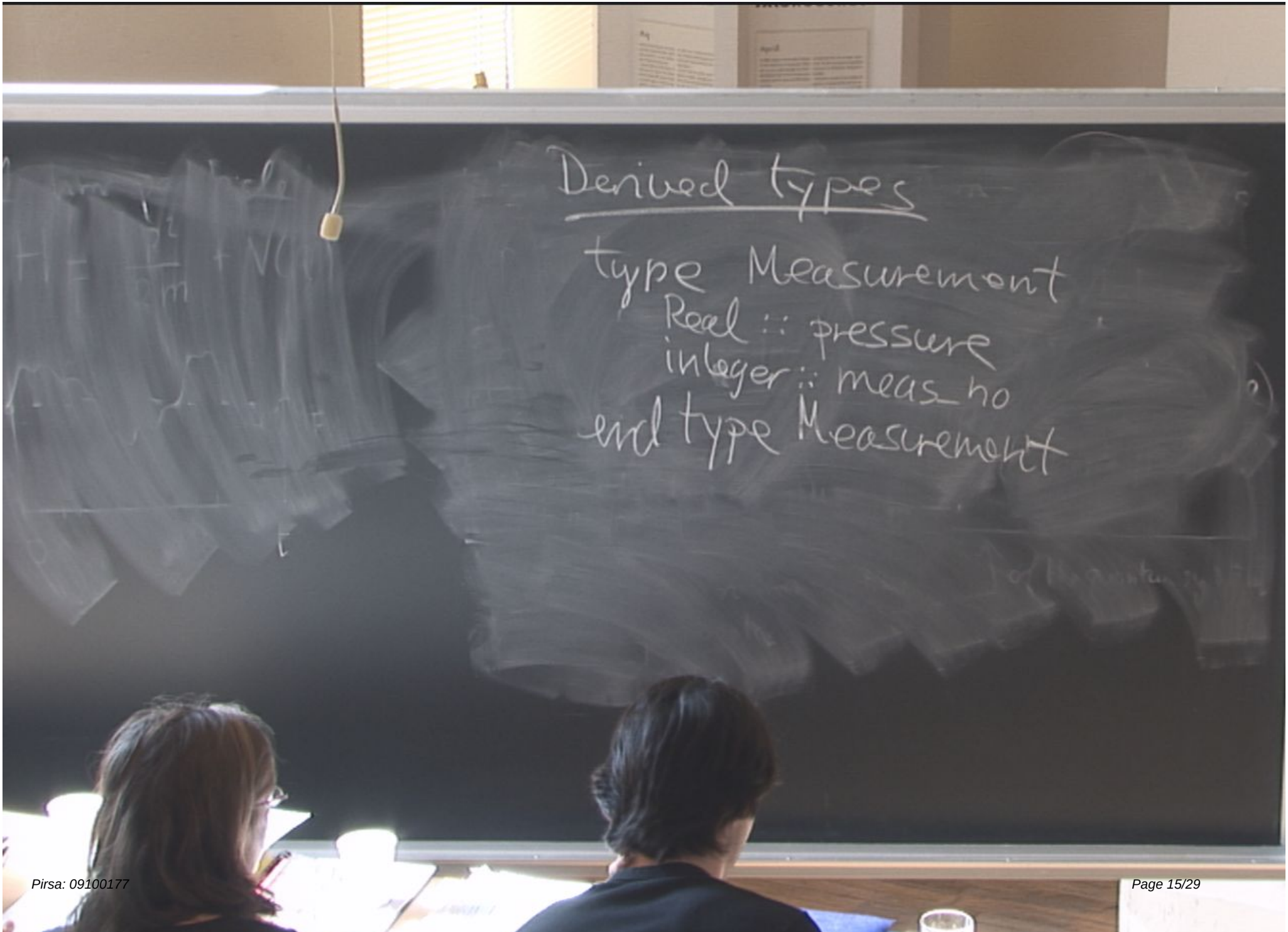

Declarations
Attribute
Real, parameter :: X = 1.0
Real :: Y = 2.0
Integer :: i
Character (len=3)
← Constant
← Initialized
Start Up

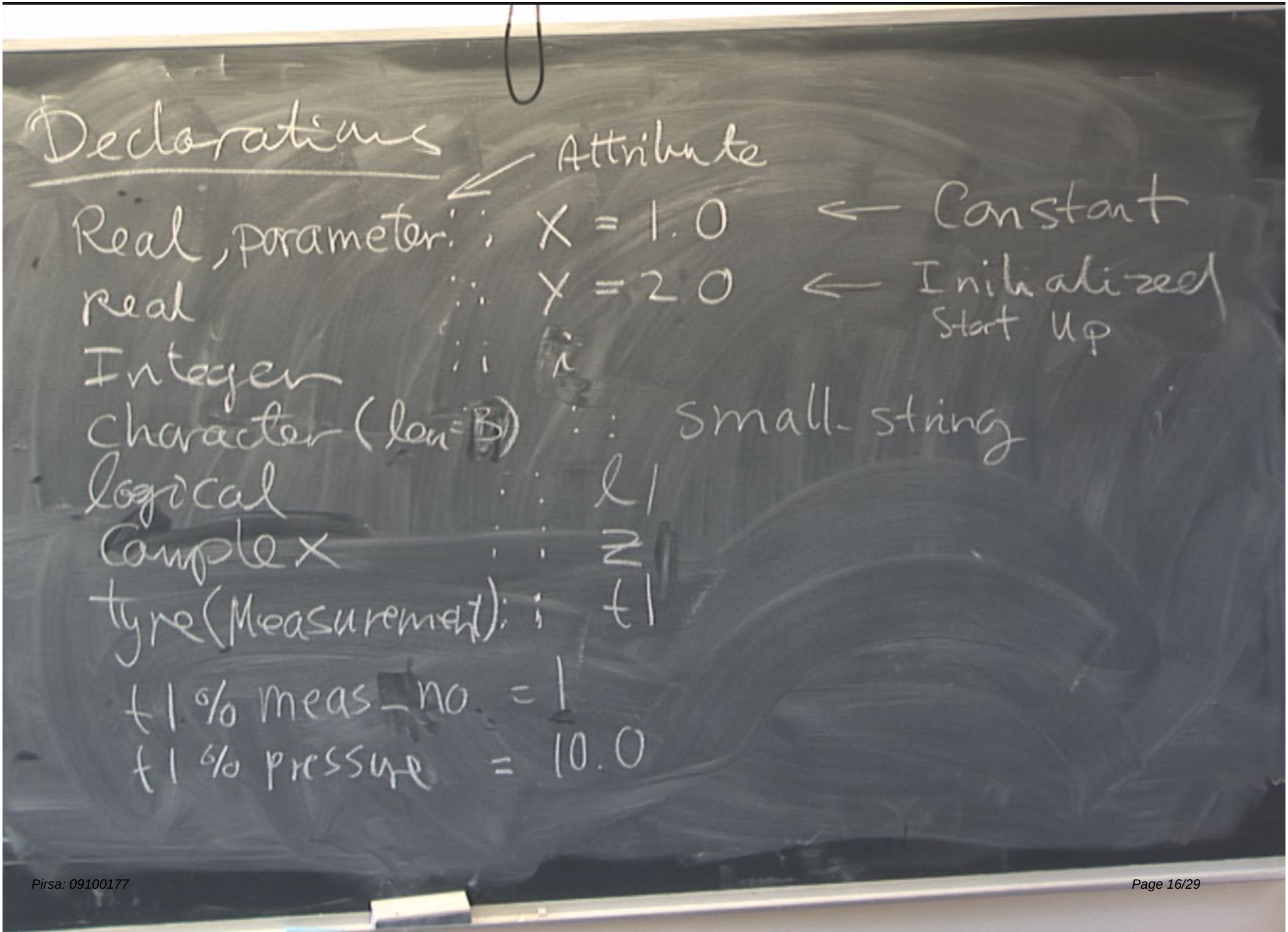
Declarations

Attribute

```
Real, parameter :: X = 1.0   <- Constant
Real            :: Y = 2.0   <- Initialized Start Up
Integer         :: i
Character(len=8) :: small_string
logical         :: l1
Complex         :: z

z = (1.0, 2.0)
```


Derived types
type Measuremont
Real :: pressure
integer :: meas_no
end type Measurement


Declarations
Attribute
Real, parameter :: X = 1.0
Constant
real :: Y = 2.0
Initialized
Start Up
Integer :: i
character (len=8) :: small_string
logical :: l1
Complex :: z
type(Measurement) :: t1
t1 % meas_no = 1
t1 % pressure = 10.0

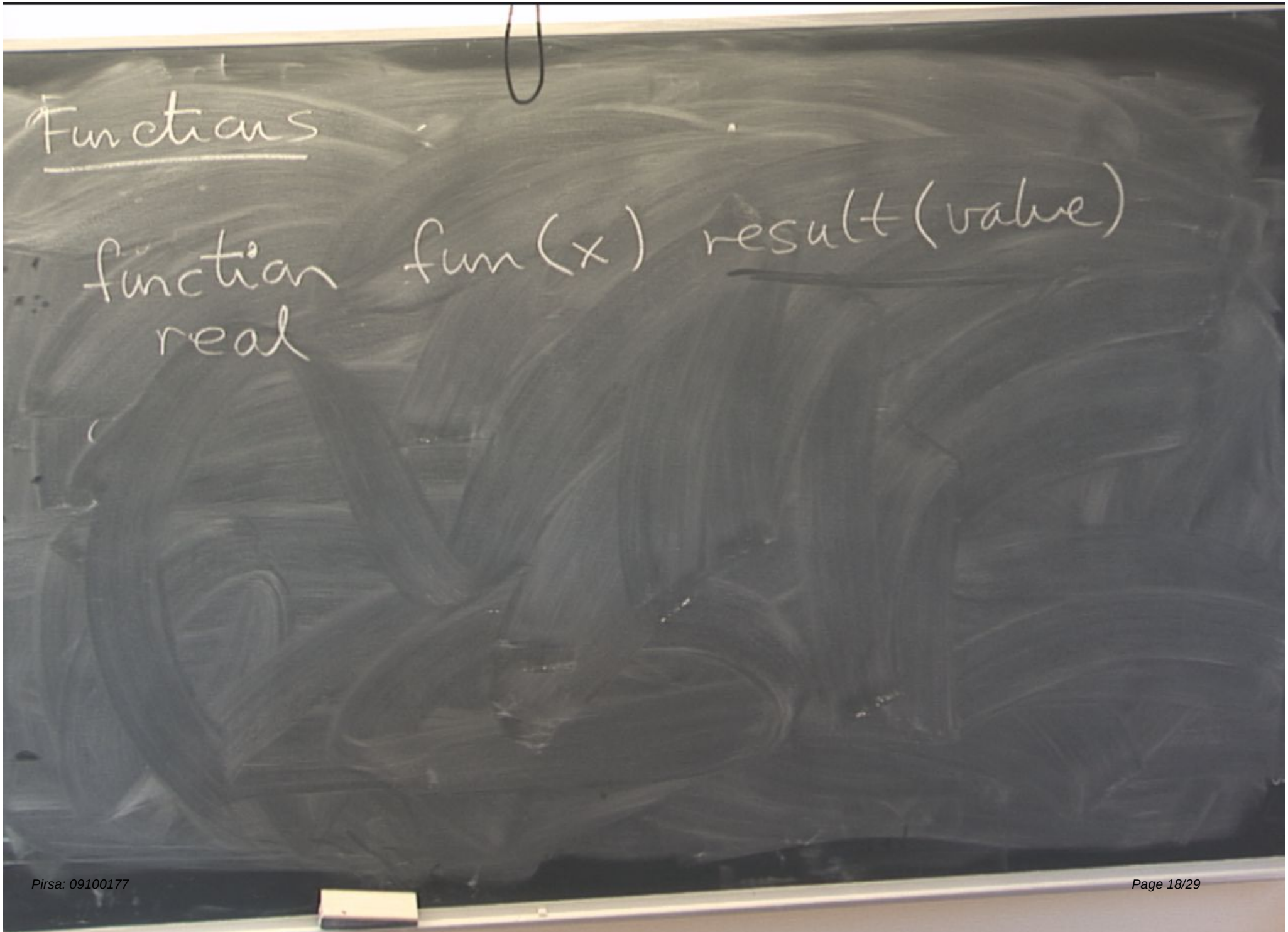
The Simplest fortran "Program"

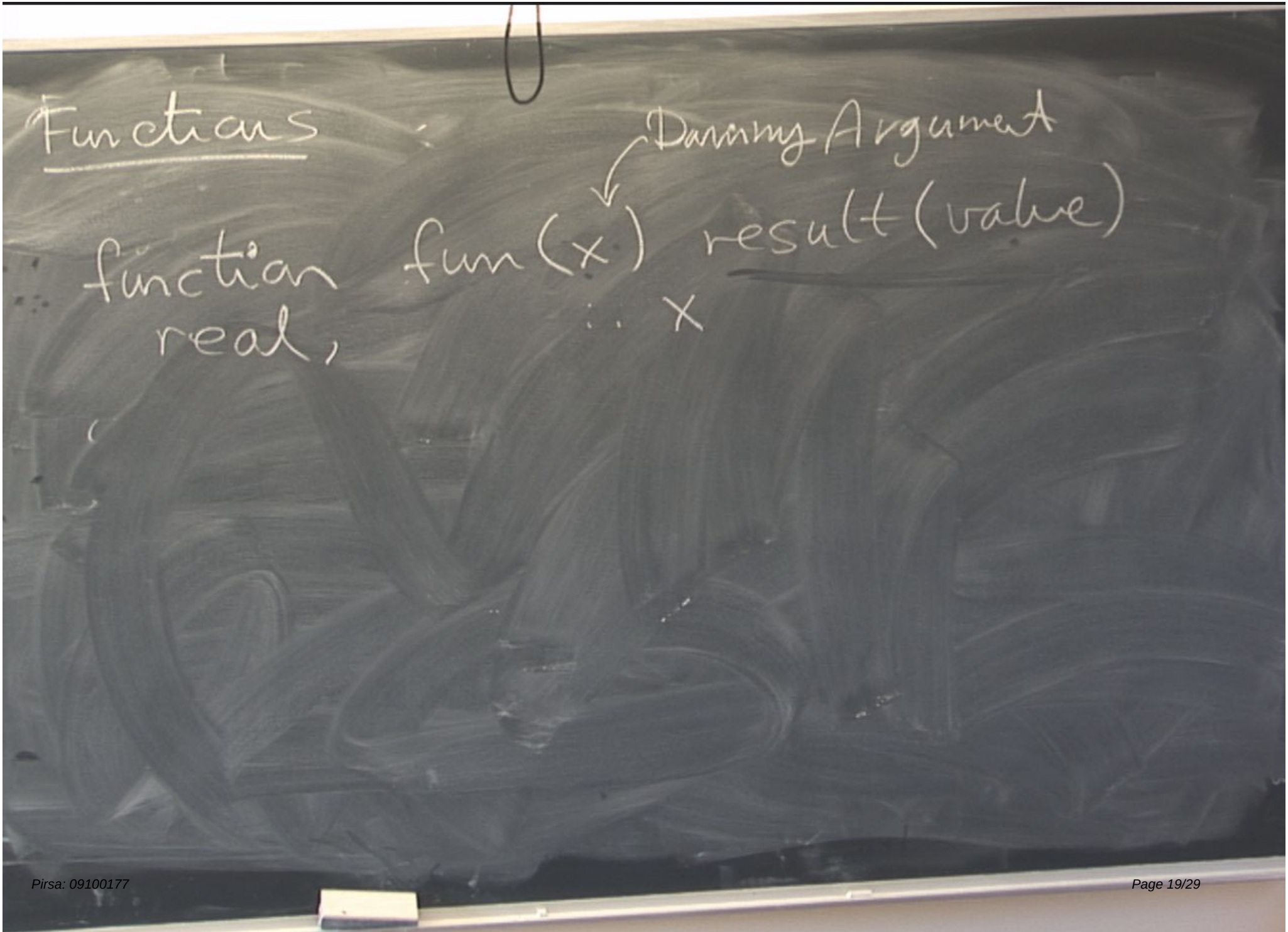
```
Program Circle
  Implicit NONE
  ! This program does
  Real :: x, y
  Read(*,*) x
  y = sqrt(1 - x*x)
  write(*,*) y
end Program Circle
```

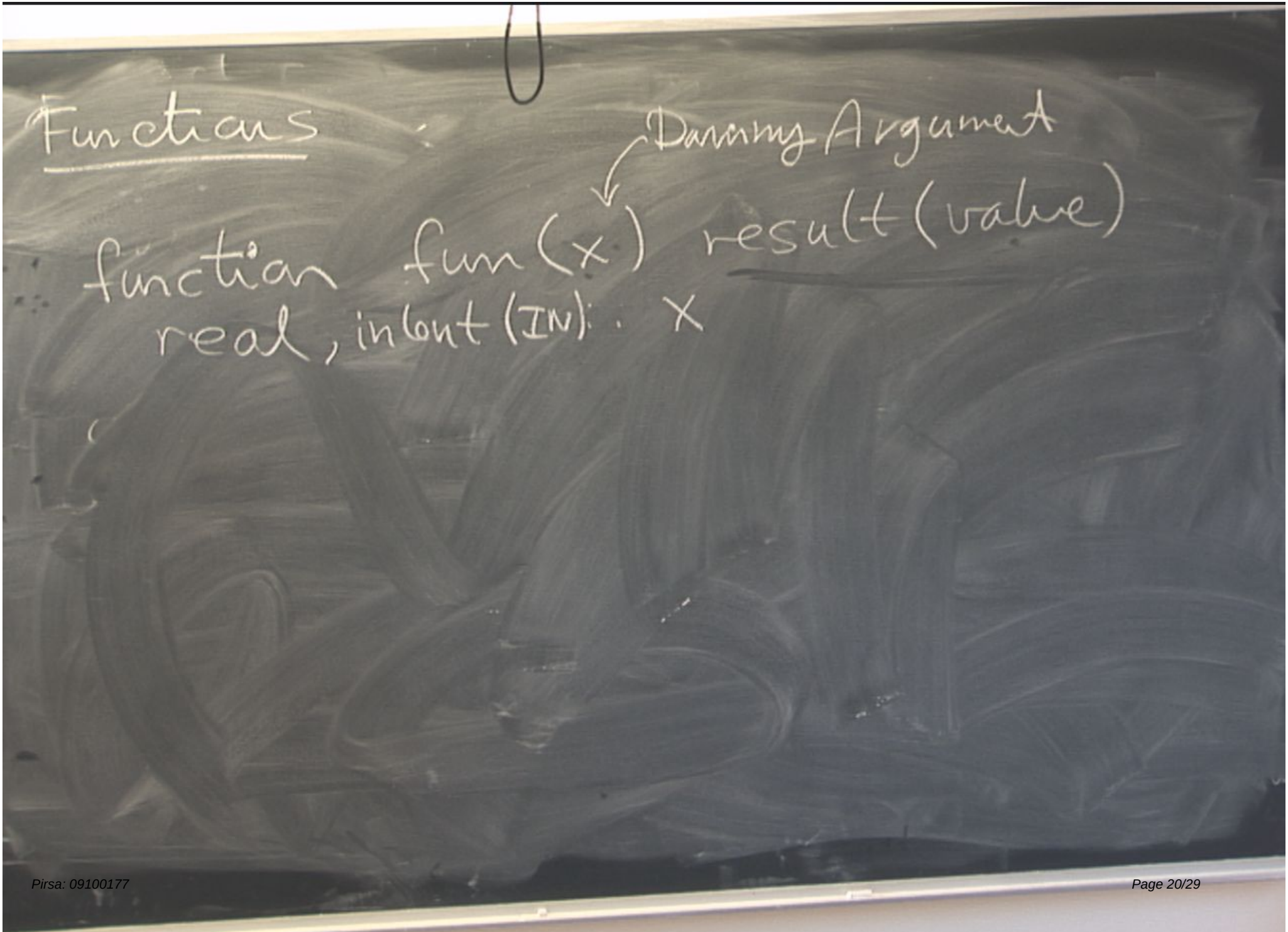
declaration part

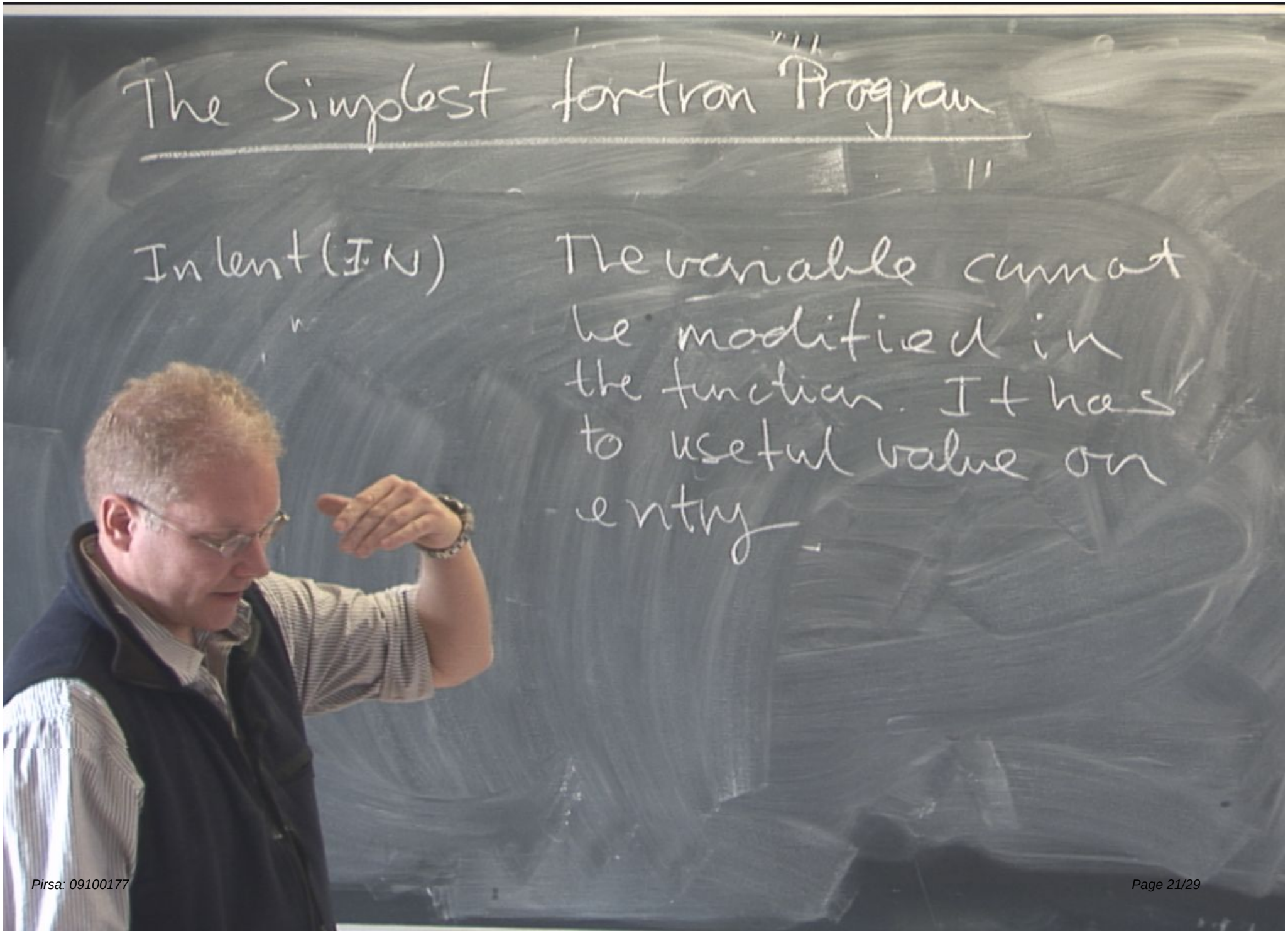
type

Statements


Functions
function fun(x) result(value)
real


Functions
Dummy Argument
function fun(x) result(value)
real, :: x

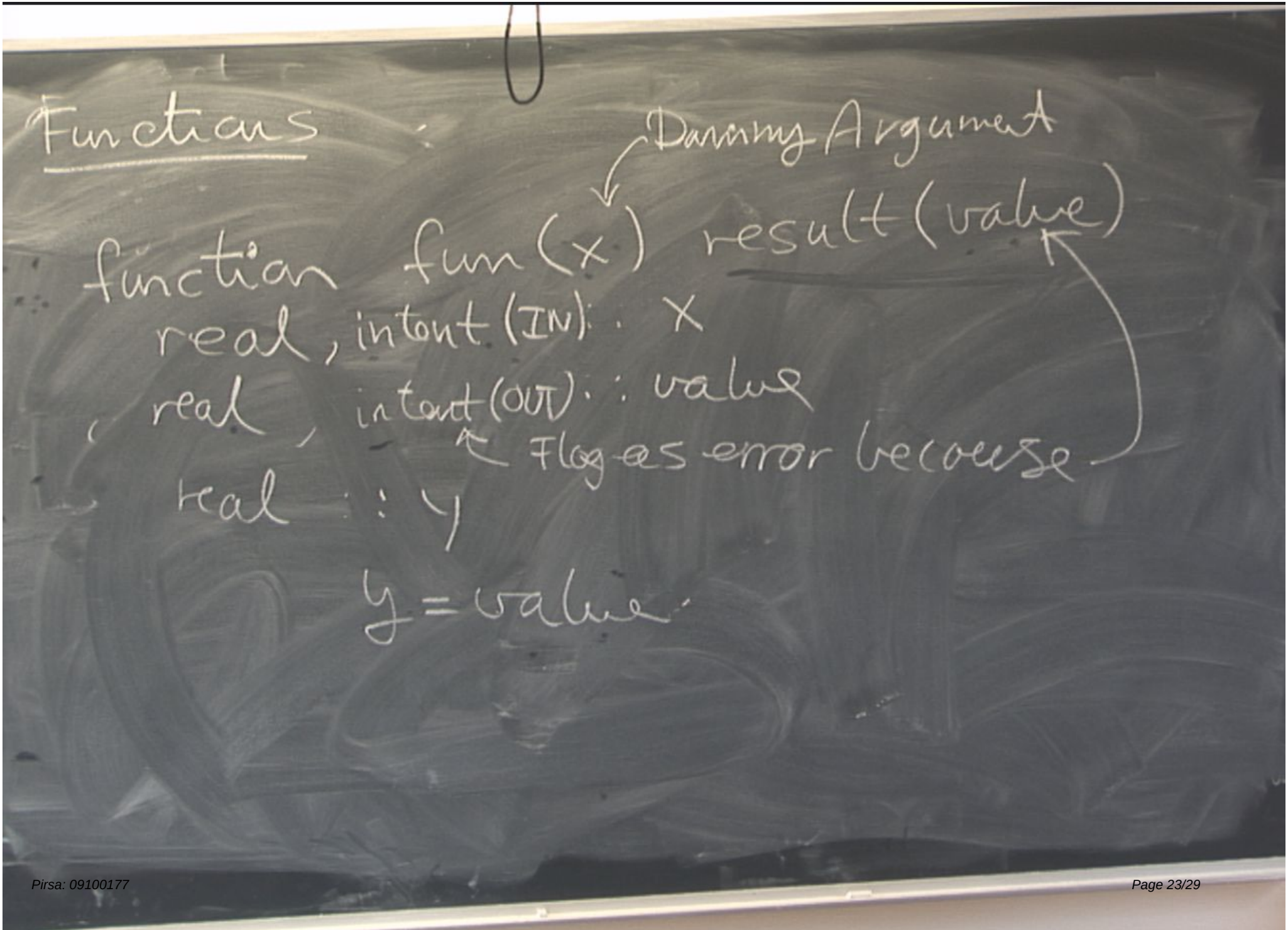

Functions
Dummy Argument
function fun(x) result(value)
real, intent(IN) :: x


The Simplest fortran "Program"
Intent(IN)
The variable cannot be modified in the function. It has to useful value on entry.

The Simplest fortran "Program"

Intent(IN) — The variable cannot be modified in the function. It has to useful value on entry

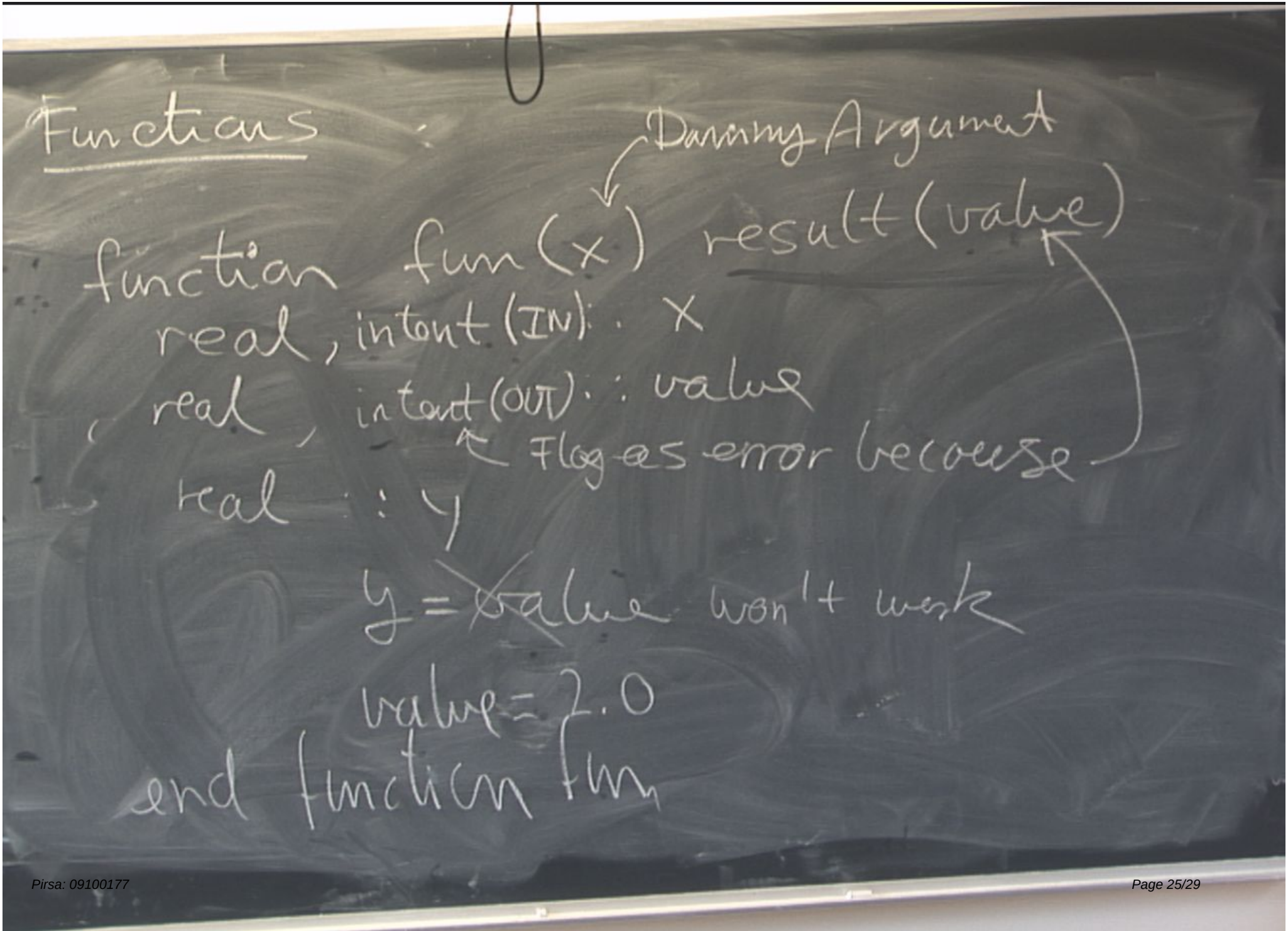
Intent(OUT) — The variable is undefined on entry

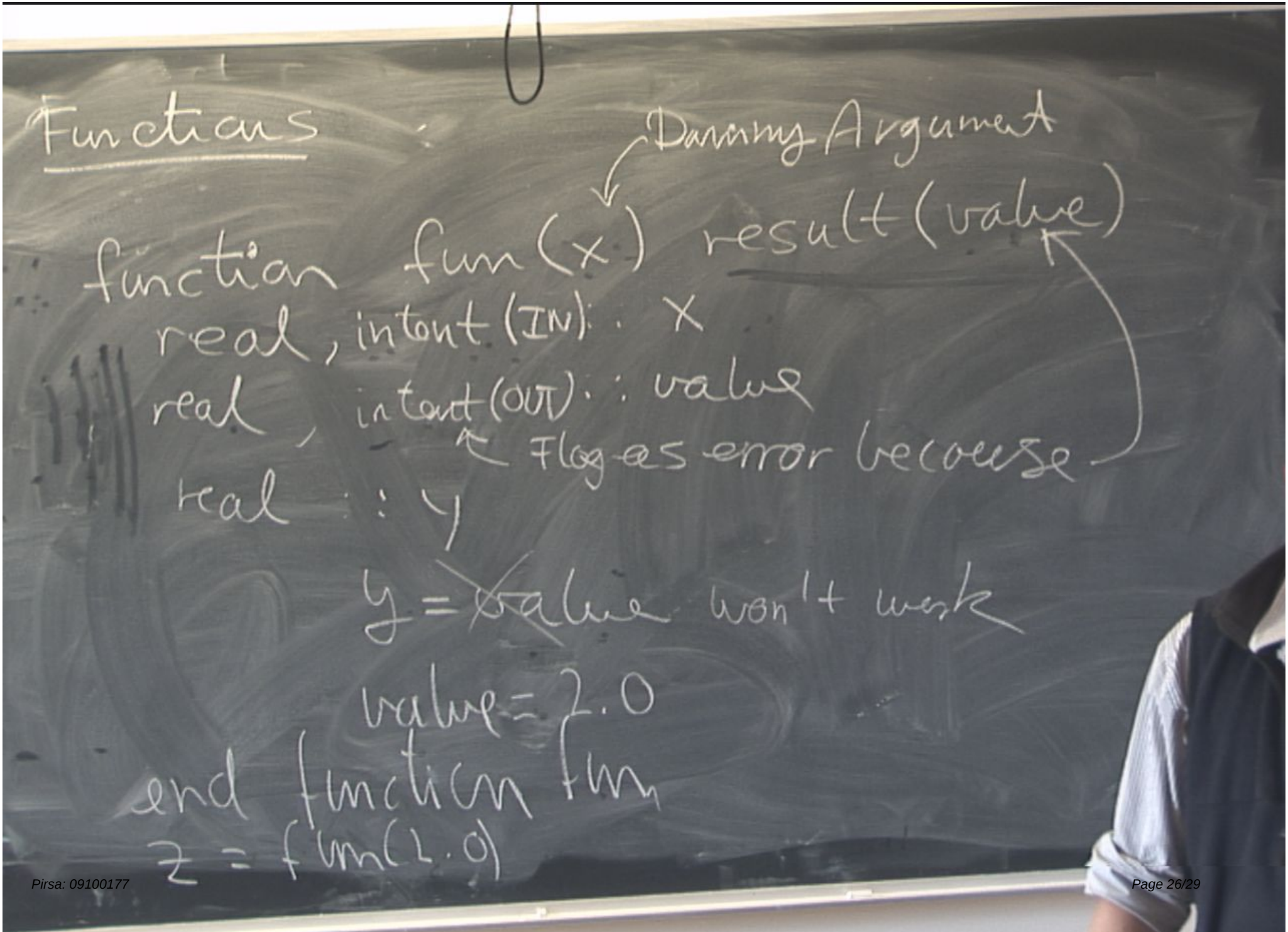

Functions
Dummy Argument
function fun(x) result(value)
real, intent(IN) :: x
real, intent(OUT) :: value
Flag as error because
real :: y
y = value

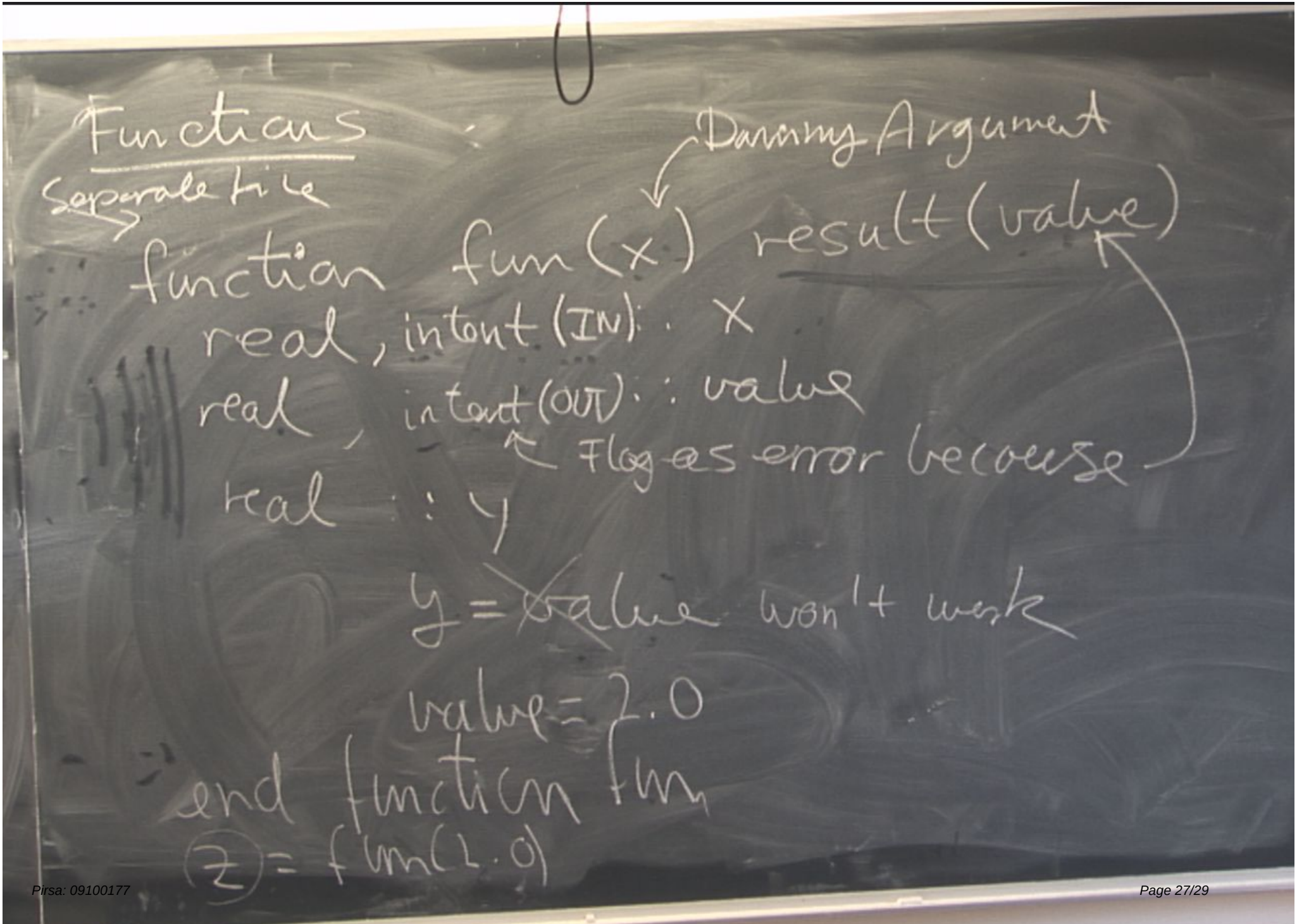
The Simplest fortran "Program"

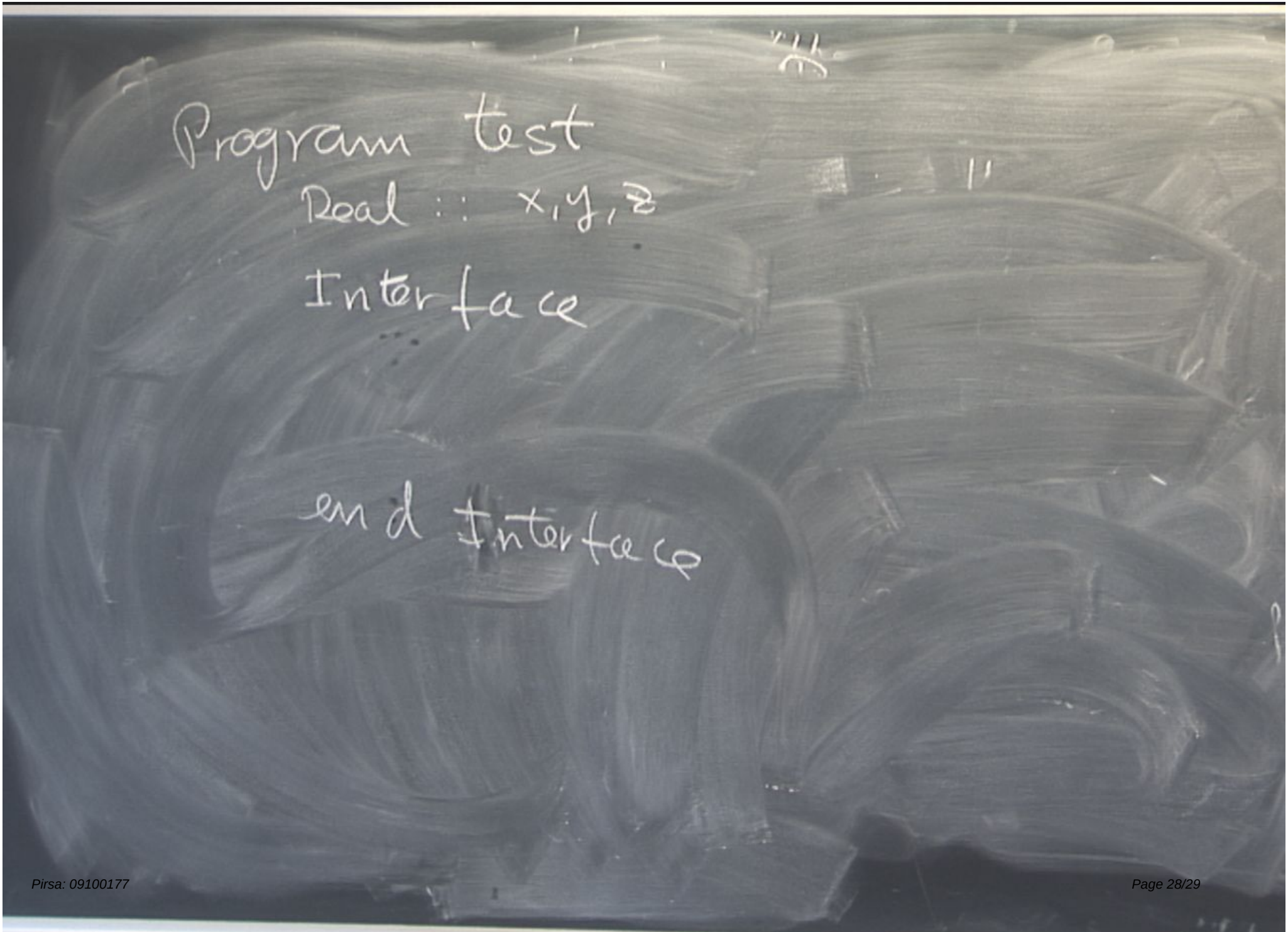
Intent(IN) — The variable cannot be modified in the function. It has to useful value on entry

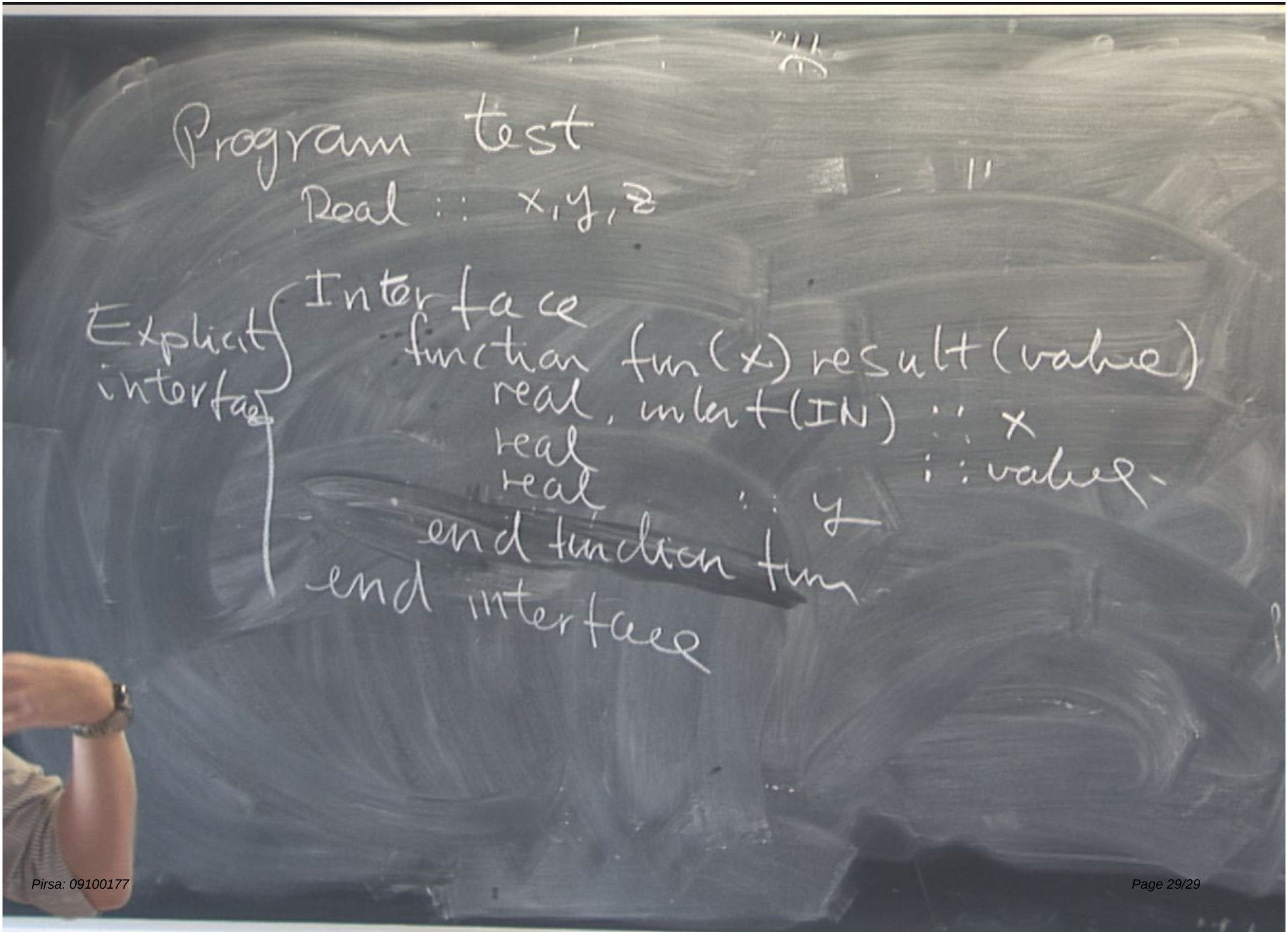
Intent(OUT) — The variable is undefined on entry. It has to be assigned a value before exit.


Functions
Dummy Argument
function fun(x) result(value)
real, intent(IN) :: x
real, intent(OUT) :: value
Flag as error because
real :: y
y = value won't work
value = 2.0
end function fun


Functions
Dummy Argument
function fun(x) result(value)
real, intent(IN) :: x
real, intent(OUT) :: value
← Flag as error because
real :: y
y = value won't work
value = 2.0
end function fun
z = fun(2.0)


Functions
Separate file
Dummy Argument
function fun(x) result(value)
real, intent(IN) :: x
real, intent(OUT) :: value
← Flag as error because
real :: y
y = value won't work
value = 2.0
end function fun
z = fun(2.0)


Program test
Real :: x, y, z
Interface
end Interface


Program test
Real :: x, y, z
Explicit interface
Interface
function fun(x) result(value)
real, intent(IN) :: x
real :: value
real :: y
end function fun
end interface