

The Assignment Statement:

The assignment statement has a form of:

variable = expression

- If the type of **variable** and **expression** are identical, the result is saved to **variable**.
- If the type of **variable** and **expression** are not identical, the result of **expression** is converted to the type of **variable**.
 - If the type of the variable is **INTEGER** while the type of the result is **REAL**, the fractional part, including the decimal point, is removed making it an integer result.
 - If the type of the variable is **REAL** while the type of the result is **INTEGER**, then a decimal point is appended to the integer making it a real number.

EX:

```
INTEGER :: Total, Amount
INTEGER :: Unit = 5

Amount = 100.99
Total = Unit * Amount
```

```
REAL, PARAMETER :: PI = 3.1415926
REAL :: Area
INTEGER :: Radius

Radius = 5
Area = (Radius ** 2) * PI
```

Input and Output

- On most computer systems the user can tell the program what values to perform a calculation upon by typing these at a keyboard. This is known as **input** and the values are assigned to the correct variables using the **READ statement**

The READ statement has the following forms:

- READ*, var1, var2, ..., varn**
- READ*,**

The following example reads in four values into variables **Factor, N, Multiple** and **tolerance** in this order

INTEGER :: Factor, N

REAL :: Multiple, tolerance

READ *, Factor, N, Multiple, tolerance

Example:

INTEGER :: P, Q, R, S

READ*, P, Q

READ*,

READ*, R, S

If the input lines are

100 200 300

400 500 600

700 800 900

The user will also wish to know the results generated by the program and this will usually be displayed on a screen using the **PRINT or WRITE statement** - this is known as **output**.

- The **PRINT** statement has the following forms:

```
Print*,  exp1, exp2, ..., expn
Print*,
```

Example:

```
INTEGER :: Target
```

```
REAL    :: Angle, Distance
```

```
CHARACTER(LEN=*), PARAMETER :: Time = "The time to hit target "  
& IS  = " is " & UNIT = " sec."
```

```
Target  = 10
```

```
Angle   = 20.0
```

```
Distance = 1350.0
```

```
Print*, 'Angle   = ', Angle
```

```
Print*, 'Distance = ', Distance
```

```
Print*,
```

```
Print*, Time, Target, IS, Angle * Distance, UNIT
```

This example may produce the following result:

```
Angle      = 20.0
```

```
Distance = 1350.0
```

```
The time to hit target 10 is 27000sec.
```

Program Structure

- Your program should have the following form:

```
PROGRAM      program-name  
  IMPLICIT   NONE  
  [specification part]  
  [execution part]  
  [subprogram part]  
END PROGRAM program-name
```