Assignment & Initialization

Assignment :- A basic operation for changing the binding of a value to the data object.

Languages like C, Lisp and many more

Assignment also returns a value which is a data object containing a copy of the value assigned.

In Pascal

assignment (:=) : integer, x integer2 -> void

Value of integer2 is copied in integer1.

In C,

assignment (=) : integer, x integer2 -> integer3

With this action: Set the value contained in data object integer1, to be a copy of the value contained in the data object integer2, and also create and return a new data object integer3, containing a copy of new value of integer2.
Two concepts through which we can define assignment

L-value: Location for an object
R-value: Content at that location.

Using L-value and R-value gives a more concise way to describe expression semantics.

Eg In case of Integer :-

\[ A = B \]

In this copying the value of variable B to variable A.

**ie assign to the lvalue of A the r-value of B**

In case of Pointer

\[ A = B \]

In this A & B are pointers variables. If B is a pointer then B's r-value is the l-value of some other data object. This assignment then means,

"Make the r-value of A refer to the same data object as the r-value of B"

Thus, the assignment \[ A = B \] means "Assign a copy of the pointer stored in variable B to variable A".

Numeric Assignment

\[
\begin{align*}
A &: 7.8 \\
B &: 0.4 \\
\end{align*}
\]

\[
\begin{align*}
A &: 0.4 \\
B &: 0.4 \\
\end{align*}
\]
Copy Value: (Pascal)

A := B

Before

After

Pointer Assignment
In C

Two Views of Assignment:

Copy Pointer: (SNOBOL)

A = B (Ptr to value of variable B assigned to variable A)