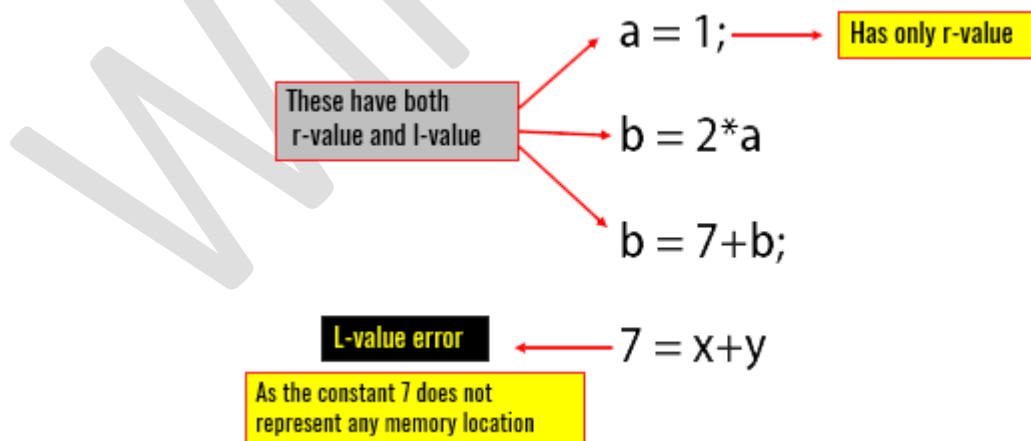## Parameter Passing in Compiler Design

- The communication medium among procedures is known as parameter passing. The values of the variables from a calling procedure are transferred to the called procedure by some mechanism.

R- value

- The value of an expression is called its r-value. The value contained in a single variable also becomes an r-value if its appear on the right side of the assignment operator.

- R-value can always be assigned to some other variable.

L-value

- The location of the memory(address) where the expression is stored is known as the l-value of that expression.

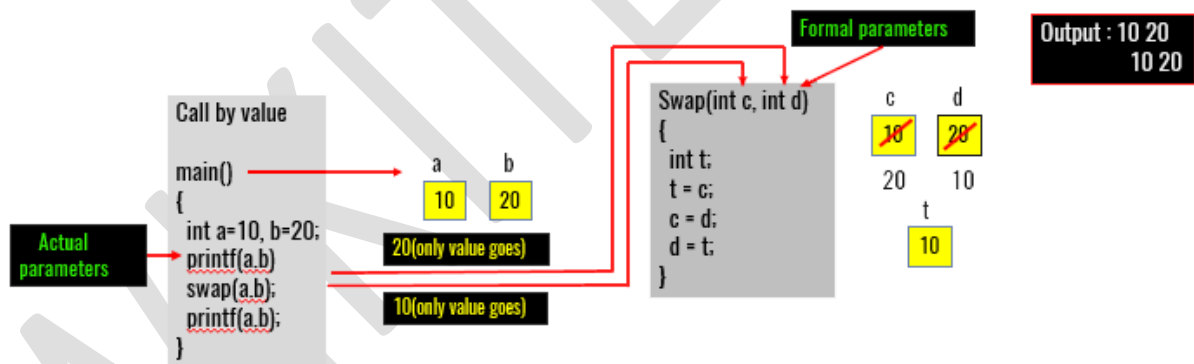- It always appears on the left side if the assignment operator.

$$a = 1; \longrightarrow \boxed{\text{Has only r-value}}$$

$$\boxed{\begin{array}{c}\text{These have both}\\ \text{r-value and l-value}\end{array}} \longrightarrow b = 2*a$$

$$b = 7+b;$$

$$\boxed{\text{L-value error}} \longleftarrow 7 = x+y$$

$$\boxed{\begin{array}{l}\text{As the constant 7 does not}\\ \text{represent any memory location}\end{array}}$$

Different ways of passing the parameters to the procedure

- Call by Value

- Call by reference

- Copy restore

- Call by name

Call by Value

- In call by value the calling procedure pass the r-value of the actual parameters and the compiler puts that into called procedure's activation record.

- Formal parameters hold the values passed by the calling procedure, thus any changes made in the formal parameters does not affect the actual parameters.

For More Details Click Here:

https://www.wikitechy.com/tutorials/compiler-design/parameter-passing-in-compiler-design