Reference Variable

First, let us understand the basics. How does the operating system (OS) display the value of variables? How are assignment operations such as ‘x=y’ executed during run time? A detailed answer to these questions is beyond the scope of this course.

A brief study is, nevertheless, possible and necessary for a good understanding of reference variables. What follows is a simplified and tailored explanation.

The OS maintains the addresses of each variable as it allocates memory for them during run time. In order to access the value of a variable, the OS first finds the address of the variable and then transfers control to the byte whose address matches that of the variable.

Suppose the following statement is executed (‘x’ and ‘y’ are integer type variables).

```
x = y;
```

The steps followed are:

1. The OS first finds the address of ‘y’.
2. The OS transfers control to the byte whose address matches this address.
3. The OS reads the value from the block of four bytes that starts with this byte (most C++ compilers cause integer-type variables to occupy four bytes during run time and we will accept this value for our purpose).
4. The OS pushes the read value into a temporary stack.
5. The OS finds the address of ‘x’.
6. The OS transfers control to the byte whose address matches this address.
7. The OS copies the value from the stack, where it had put it earlier, into the block of four bytes that starts with the byte whose address it has found above (address of ‘x’).

Notice that addresses of the variables on the left as well as on the right of the assignment operator are determined. However, the value of the right-hand operand is also determined. The expression on the right must be capable of being evaluated to a value. This is an important point and must be borne in mind. It will enable us to understand a number of concepts later.

Especially, you must remember that the expression on the left of the assignment operator must be capable of being evaluated to a valid address at which data can be written.

Now, let us study **reference variables**. A reference variable is nothing but a reference for an existing variable. It shares the memory location with an existing variable.

The syntax for declaring a reference variable is as follows:

```
<data type> & <ref-variable-name> = <existing-variable-name>;
```

For example, if ‘x’ is an existing integer-type variable and we want to declare iRef as a reference to it the statement is as follows:

```
int & iRef=x;
```

iRef is a reference to ‘x’. This means that although iRef and ‘x’ have separate entries in the OS, their addresses are actually the same!

Thus, a change in the value of ‘x’ will naturally reflect in iRef and vice versa.