

# Problem Solving Through programming in C

Course Code:ONL1001

## Assignment Operators

Ms. Shubhra dwivedi

Assistant Professor

School - SCOPE

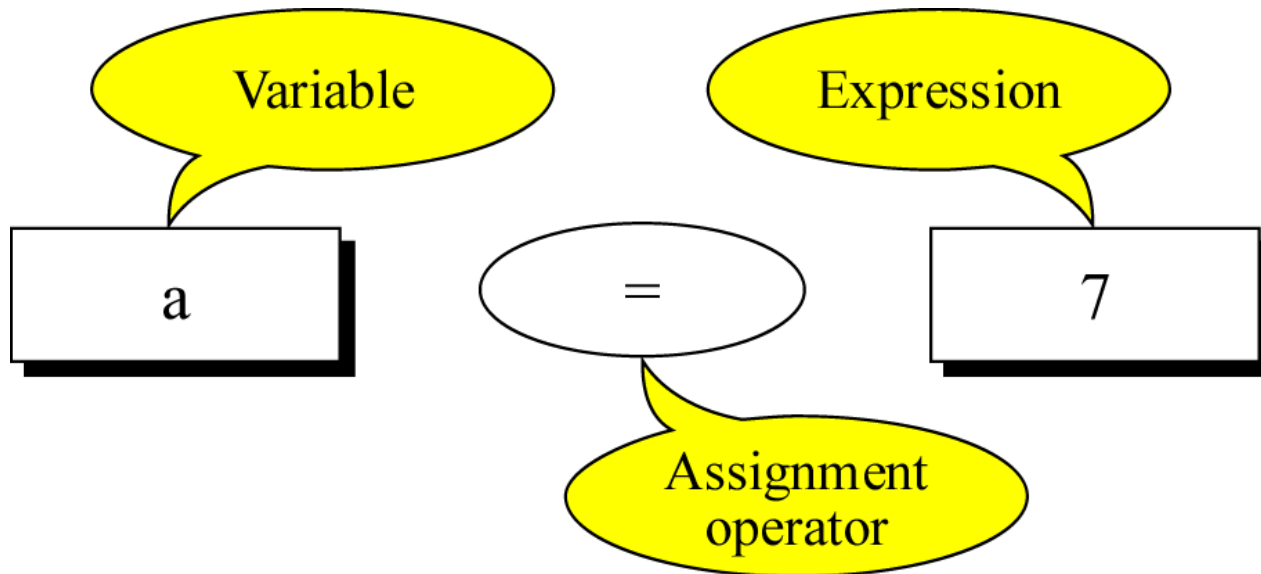
VIT-AP Amaravati

Shubhra.d@vitap.ac.in

# Assignment operator: =



- Binary operator used to assign a value to a variable.




# Difference between = and ==


- Although = and == seem to be same, but they are quite different from each other.
- = is assignment operator while == is the equality operator.
- = is used to assign values whereas == is used for comparing values.
- Take two examples.

For example:

- `x = 5;`
- `x == 5;`
- By writing `x = 5`, we assign a value of 5 to x, whereas by writing `x == 5`, we check if the value of x is 5 or not.

For example:

 = Sam (You are Sam)

 == Sam (Are you Sam?)

# Special assignment operators

- write `a += b;` instead of `a = a + b;`
- write `a -= b;` instead of `a = a - b;`
- write `a *= b;` instead of `a = a * b;`
- write `a /= b;` instead of `a = a / b;`
- write `a %= b;` instead of `a = a % b;`

```

#include <stdio.h>
int main()
{
    int a = 25;
    int b = 8;
    a += b;
    printf("%d\n",a);
    a -=b;
    printf("%d\n",a);
    a *=b;
    printf("%d\n",a);
    a /=b;
    printf("%d\n",a);
    a %=b;
    printf("%d\n",a);
    return 0;
}

```

Output

33  
25  
200  
25  
1

Initially 'a' is 25 and 'b' is 8.

a += b; → It means a = a+b. Now, 'a' is a+b i.e. 33. (a = a+b)

a -= b → Now, 'a' is 33. So, a -= b (a = a-b) is 25. So, new 'a' is 25. Similarly, the values will be evaluated for the next three cases.

# Special assignment operators

- Increment, decrement operators: ++, --
  - Instead of `a = a + 1` you can write `a++` or `++a`
  - Instead of `a = a - 1` you can write `a--` or `--a`
- What is the difference?

## *post-increment*

```
num = 10;  
ans = num++;
```

First assign num to ans,  
then increment num.  
In the end,

num is 11  
ans is 10

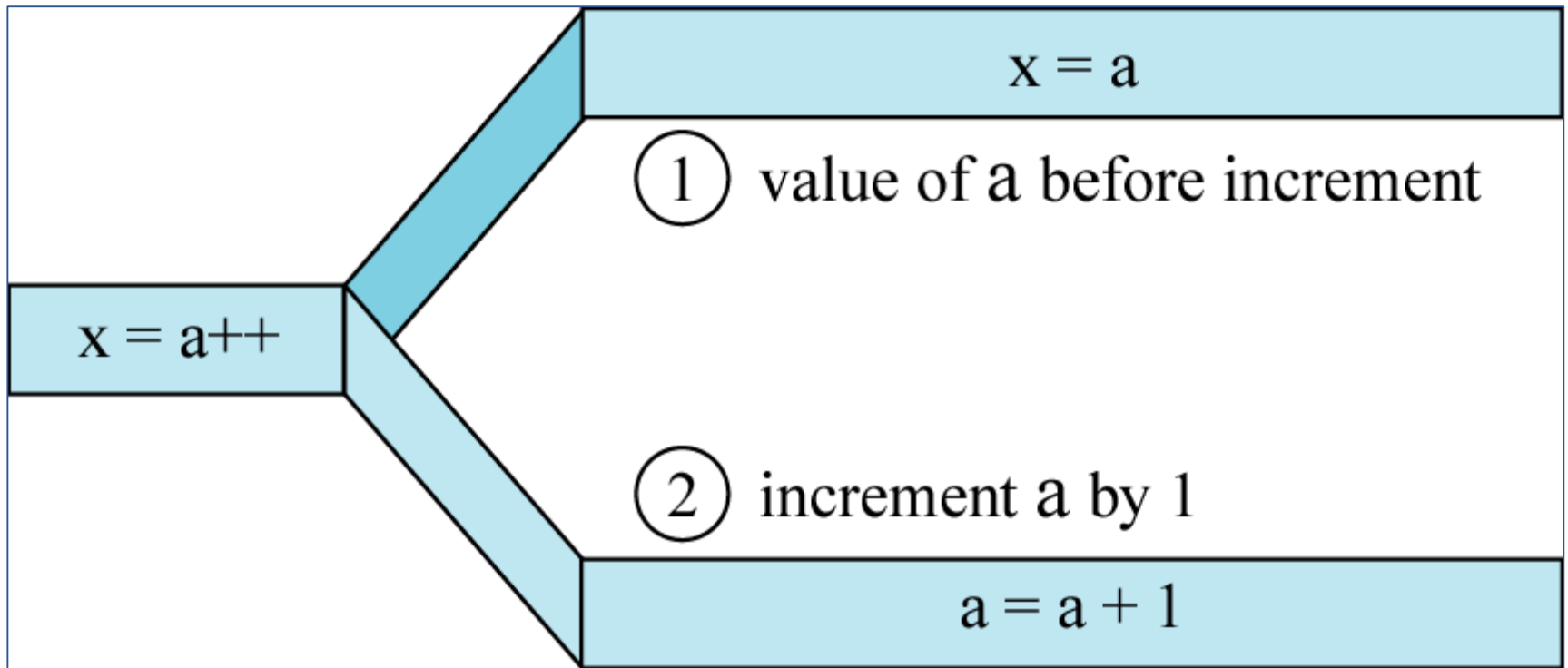
## *pre-increment*

```
num = 10;  
ans = ++num;
```

First increment num,  
then assign num to ans.  
In the end,

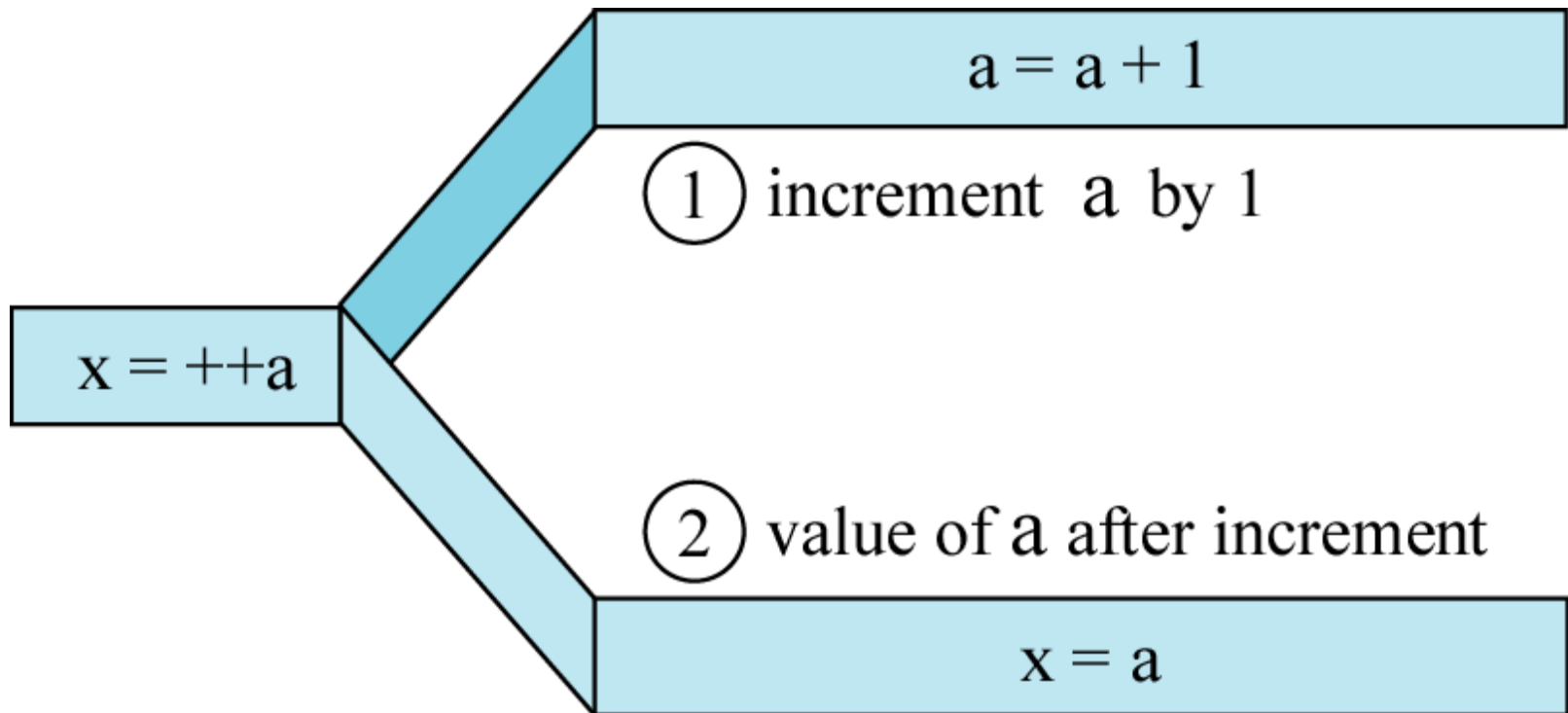
num is 11  
ans is 11

## Result of postfix Increment





## Result of Prefix Increment



```
#include <stdio.h>
int main()
{
    int a = 15, b = 15, c = 15, d = 15;
    printf("\nvalue of a++ = %d", a++);
    printf("\nvalue of ++b = %d", ++b);
    printf("\nvalue of c-- = %d", c--);
    printf("\nvalue of --d = %d", --d);
    return 0;
}
```

### Output

```
value of a++ = 15
value of ++b = 16
value of c-- = 15
value of --d = 14
```

In a++ 15 is printed. Then the value of a will become 16. And in ++b, value is first increased to 16 and then printed. Similar with c-- and --d.