

In C, the increment operator '++' is used to increase the value of a variable by 1. There are two forms of the increment operator:

1. Pre-increment
2. Post-increment.

1. Pre-increment (++x):

- Pre-increment increments the value of the variable before it is used in an expression.
- The syntax is ++x.
- The value of x is incremented by 1, and the updated value is used in the expression.

Example:

```
#include <stdio.h>

int main() {
    int x = 5;
    // Pre-increment: increments the value of x before using it
    int y = ++x;
    printf("x: %d\n", x); // Output: x: 6
    printf("y: %d\n", y); // Output: y: 6
    return 0;
}
```

In this program, we have a variable x initialized to 5. Using the pre-increment operator ++x, the value of x is incremented by 1 before it is assigned to the variable y. Therefore, both x and y will have a value of 6.

The output of this program will be:

```
x: 6  
y: 6
```

2. Post-increment (x++):

- Post-increment increments the value of the variable after it is used in an expression.
- The syntax is x++.
- The current value of x is used in the expression, and then the value of x is incremented by 1.

Example:

```
#include <stdio.h>  
  
int main() {  
    int x = 5;  
    // Post-increment: increments the value of x after using it  
    int y = x++;  
    printf("x: %d\n", x); // Output: x: 6  
    printf("y: %d\n", y); // Output: y: 5  
    return 0;  
}
```

Here, the post-increment operator x++ is used. It assigns the current value of x to y and then increments the value of x by 1. As a result, x becomes 6 while y retains the original value of 5.

The output of this program will be:

```
x: 6  
y: 5
```

Difference between Pre and Post Increment

	Pre-increment (++x)	Post-increment (x++)
1	Increments the value of x before use	Increments the value of x after use
2	Uses the updated value of x	Uses the current value of x
3	Returns the updated value of x	Returns the current value of x
4	Example:	Example:
	"`c	"`c
	int x = 5;	int x = 5;
	int y = ++x; // y = 6, x = 6	int y = x++; // y = 5, x = 6

PRE AND POST INCREMENT PRACTICE PROBLEMS

1. What is the value of “x” after the following code snippet executes?

```
int x = 5;  
x++;
```

Explanation: The initial value of “x” is 5. The post-increment operator (x++) is applied to “x”, which increments the value of “x” by 1. After the operation, the value of “x” becomes 6.

2. What is the value of “y” after the following code snippet executes?

```
int y = 3;  
int x = y++;
```

Explanation: The initial value of “y” is 3. The post-increment operator (y++) is applied to “y” during the assignment to “x”. Therefore, the original value of “y” (3) is assigned to “x”, and then “y” is incremented by 1. After the operation, the value of “y” becomes 4.

3. What is the value of “x” after the following code snippet executes?

```
int x = 2;  
int y = x++ + ++x;
```

Explanation: The initial value of “x” is 2. In the expression `x++ + ++x`, the post-increment operator (`x++`) is applied to the first occurrence of “x” and the pre-increment operator (`++x`) is applied to the second occurrence of “x”. The post-increment operator returns the original value of “x” (2) and then increments “x” by 1. The pre-increment operator increments “x” by 1 and returns the updated value (4). Therefore, the expression evaluates to `2 + 4`, resulting in 6. After the operation, the value of “x” becomes 4.

4. What is the output of the following code snippet?

```
int x = 2;  
System.out.println(x++);  
System.out.println(x);
```

Explanation: The initial value of “x” is 2. In the first `println` statement, the post-increment operator (`x++`) is applied to “x”. The original value of “x” (2) is printed, and then “x” is incremented by 1. Therefore, the output is 2. In the second `println` statement, the updated value of “x” (3) is printed. Therefore, the output is 3.

5. What is the value of “y” after the following code snippet executes?

```
int y = 5;  
int x = ++y + y++;
```

Explanation: The initial value of “y” is 5. In the expression `++y + y++`, the pre-increment operator (`++y`) is applied to the first occurrence of “y” and the post-increment operator (`y++`) is applied to the second occurrence of “y”. The pre-increment operator increments “y” by 1 and returns the updated value (6). The post-increment operator returns the original value of “y” (6) and then increments “y” by 1. Therefore, the expression evaluates to `6 + 6`, resulting in 12. After the operation, the value of “y” becomes 7.

6. What is the value of “x” after the following code snippet executes?

```
int x = 3;  
x = ++x + x++;
```

Explanation: The initial value of “x” is 3. In the expression `++x + x++`, the pre-increment

operator ($++x$) is applied to the first occurrence of "x" and the post-increment operator ($x++$) is applied to the second occurrence of "x". The pre-increment operator increments "x" by 1 and returns the updated value (4). The post-increment operator returns the original value of "x" (4) and then increments "x" by 1. Therefore, the expression evaluates to $4 + 4$, resulting in 8. After the operation, the value of "x" becomes 4.

7. What is the output of the following code snippet?

```
int x = 2;  
System.out.println(++x + x++);
```

Explanation: The initial value of "x" is 2. In the println statement, the pre-increment operator ($++x$) is applied to the first occurrence of "x" and the post-increment operator ($x++$) is applied to the second occurrence of "x". The pre-increment operator increments "x" by 1 and returns the updated value (3). The post-increment operator returns the original value of "x" (3) and then increments "x" by 1. Therefore, the expression evaluates to $3 + 3$, resulting in 6. The output will be 6.

8. What is the value of "y" after the following code snippet

executes?

```
int x = 4;  
int y = x++ + ++x;
```

Explanation: The initial value of "x" is 4. In the expression `x++ + ++x`, the post-increment operator (`x++`) is applied to the first occurrence of "x" and the pre-increment operator (`++x`) is applied to the second occurrence of "x". The post-increment operator returns the original value of "x" (4) and then increments "x" by 1. The pre-increment operator increments "x" by 1 and returns the updated value (6). Therefore, the expression evaluates to `4 + 6`, resulting in 10. After the operation, the value of "y" becomes 10.

9. What is the value of "x" after the following code snippet executes?

```
int x = 10;  
int y = x-- - --x;
```

Explanation: The initial value of "x" is 10. In the expression `x-- - --x`, the post-decrement operator (`x--`) is applied to the first occurrence of "x" and the pre-decrement operator (`--x`) is applied to the second occurrence of "x". The post-decrement operator returns the original value of "x" (10) and then decrements "x" by 1. The pre-decrement operator decrements "x" by 1 and returns the updated value (8). Therefore, the expression evaluates to `10 - 8`,

resulting in 2. After the operation, the value of "x" becomes 8.

10. What is the output of the following code snippet?

```
int x = 5;  
System.out.println(x--);  
System.out.println(x);
```

Explanation: The initial value of "x" is 5. In the first println statement, the post-decrement operator (x--) is applied to "x". The original value of "x" (5) is printed, and then "x" is decremented by 1. Therefore, the output is 5. In the second println statement, the updated value of "x" (4) is printed. Therefore, the output is 4.

11. What is the result of the expression

```
x = 5; y = ++x;
```

- a) x = 6, y = 6
- b) x = 6, y = 5
- c) x = 5, y = 6

d) $x = 5, y = 5$

12. What is the value of “x” after the following code snippet executes ?

```
int x = 3;  
int y = x++;
```

- a) 3
 - b) 4
 - c) 5
 - d) 6
-

13. What is the value of “y” after the following code snippet executes?

```
int x = 2;  
int y = x++ + ++x;
```

- a) 4
- b) 5
- c) 6
- d) 7

14. What is the output of the following code snippet?

```
int x = 5;  
System.out.println(x++);  
System.out.println(x);
```

- a) 5, 6
 - b) 6, 6
 - c) 6, 5
 - d) 5, 5
-

15. What is the value of “y” after the following code snippet executes?

```
int x = 2;  
int y = x++ * ++x;
```

- a) 4
- b) 6
- c) 8

d) 10

16. What is the value of “x” after the following code snippet executes?

```
int x = 4;  
x = x++ + ++x;
```

- a) 7
- b) 8
- c) 9
- d) 10

17. What is the output of the following code snippet?

```
int x = 1;  
System.out.println(++x + x++);
```

- a) 4
- b) 5
- c) 6
- d) 7

18. What is the value of “y” after the following code snippet executes?

```
int x = 3;  
int y = ++x + x++;
```

- a) 7
- b) 8
- c) 9
- d) 10

19. What is the value of “x” after the following code snippet executes?

```
int x = 10;  
int y = x-- - --x;
```

- a) 8
- b) 9
- c) 10
- d) 11

20. What is the output of the following code snippet?

```
int x = 5;  
System.out.println(x--);
```

```
System.out.println(x);
```

- a) 5, 4
- b) 5, 5
- c) 4, 4
- d) 4, 5

Related Posts:

1. C prgoram to convert inch to feet
2. C program to convert KM to CM
3. C program to convert meter to centimeter
4. C program to calculate remainder, difference, division, product
5. C program to use printf() without semicolon " ; "
6. C program to swap two numbers using 2 variables
7. C program to find nth term using Arithmetic progrssion
8. C program to find sum of first n even positive numbers
9. C program to calculate sum of first n even numbers
10. C program to find nth odd number
11. C program to find sum of first n odd positive numbers
12. C program to calculate perimeter and area of a rectangle
13. C program to calculate perimeter and area of a square
14. C program to calculate Perimeter and Area of Circle
15. Function in C Programming
16. C Programming Q & A
17. Main function in C Programming Q and A
18. Void main in C Programming

19. Variables Q and A in C Programming
20. Write a C Program to find the percentage of marks ?
21. Write a c program to find age of a person ?
22. Write a c program to get table of a number
23. What is Break statement in C Programming ?
24. Write a c program to generate all combinations of 1, 2 and 3 using for loop.
25. Write a C program to print all the prime numbers between 1 to 50.
26. Write a C program to get factorial of a number ?
27. What is user defined function in C programming ?
28. Difference between C and C++ Programming ?
29. Difference between C, C++ and Java Programming
30. C program addition of numbers using pointer
31. C Syntax
32. Comments in C
33. Variables in C
34. Data types in C
35. Format specifiers in C
36. Type Conversion in C
37. Constants in C
38. Operators in C
39. Pre and Post Increment Practice Problems
40. Array in C
41. C Introduction
42. C Get Started
43. C Pointers
44. C History
45. C Program Compiling and running

46. C While loop
47. C Do While Loop
48. C For loop
49. break and continue statement
50. Control Statements in C
51. C if-else ladder
52. C if statements
53. C 2-Dimensional array
54. C String library functions
55. C Functions
56. C Functions Categories
57. C Actual Arguments
58. Write a program that prints the message "Hello, World!"
59. Write a program that asks the user to enter two numbers, and then prints the sum of those two numbers.
60. Write a program that asks the user to enter a number and then determines whether the number is even or odd.
61. Write a program that swaps the values of two variables.
62. Write a program that asks the user to enter a number and then calculates and prints its factorial.
63. Write a program that asks the user to enter a number N and then prints the first N numbers in the Fibonacci sequence
64. Write a program that swaps the values of two variables without using a temporary variable
65. Converts a number into integer, float, and string
66. Program to find the length of the string
67. Program to convert string to uppercase or lowercase

- 68. Program to prints the numbers from 1 to 10.
- 69. What is identifier expected error
- 70. Difference between static and non static methods in Java
- 71. C String Input
- 72. C Character input
- 73. C Programming Variables MCQ
- 74. Object & Classes
- 75. C Programming find the output MCQs