



Fundamentos de Programación 101

By Ernie

Ernesto José Canales Guillén

Círculos de estudio UCA

Ciclo Virtual 01/2021



Introducción a la programación en C++



Operadores aritméticos

Se utilizan para realizar operaciones matemáticas comunes.

Operator	Name	Description	Example
+	Addition	Adds together two values	$x + y$
-	Subtraction	Subtracts one value from another	$x - y$
*	Multiplication	Multiplies two values	$x * y$
/	Division	Divides one value by another	x / y
%	Modulus	Returns the division remainder	$x \% y$
++	Increment	Increases the value of a variable by 1	$++x$
--	Decrement	Decreases the value of a variable by 1	$--x$



Operadores de incremento y disminución



Operadores de incremento y disminución

- C++ proporciona el operador de incremento, `++`, que aumenta el valor de una variable en 1, y el operador de decremento, `--`, que reduce el valor de una variable en 1.
- La sintaxis del operador de incremento es:
 - Pre-incremento: `++variable`
 - Post-incremento: `variable++`
- La sintaxis del operador de decremento es:
 - Pre-decremento: `--variable`
 - Post-decremento: `variable--`



¿Cuál es la diferencia entre las formas pre y post de estos operadores?

La diferencia se hace evidente cuando la variable que utiliza estos operadores se emplea en una expresión:

Suponga que `x` es una variable `int`. Si `++x` se usa en una expresión, primero el valor de `x` es incrementado en 1, y luego el nuevo valor de `x` se usa para evaluar la expresión.

Por otro lado, si se usa `x++` en una expresión, primero se usa el valor actual de `x` en la expresión, y luego el valor de `x` se incrementa en 1.



Declaración (creación) de variables

Las variables son contenedores para almacenar valores de datos.



Declarando variables

- Variable de una sola linea.
- Variable de varias lineas.



```
1 int myNum = 15; //Single line variable  
2 int x = 5, y = 6, z = 50; //Multiple line variable
```




Constantes

Cuando no desee que otros (o usted mismo) anulen los valores de las variables existentes, use la palabra clave `const` (esto declarará la variable como "constante", lo que significa inmutable y de solo lectura).



```
1 const int myNum = 15; // myNum will always be 15  
2 myNum = 10; // error: assignment of read-only variable 'myNum'
```



C++ Identifiers

- Todas las variables de C ++ deben identificarse con nombres únicos.
- Las reglas generales para construir nombres para variables (identificadores únicos) son:
 - Los nombres pueden contener letras, dígitos y guiones bajos.
 - Los nombres deben comenzar con una letra o un guión bajo (_).
 - Los nombres distinguen entre mayúsculas y minúsculas (myVar y myvar son variables diferentes).
 - Los nombres no pueden contener espacios en blanco ni caracteres especiales como !, #, %, etc...
 - Las palabras reservadas no se pueden usar como nombres.



Naming Conventions

- Camel Case:
 - La primera letra de cada palabra está en mayúscula sin espacios ni símbolos entre palabras.
- Pascal Case:
 - Popularizado por el lenguaje de programación Pascal, este es un subconjunto de Camel Case donde la palabra comienza con mayúsculas.
- Snake Case:
 - Las palabras dentro de frases o palabras compuestas se separan con un guión bajo.
- Kebab Case:
 - Como Snake Case, pero usando guiones en su lugar.
- Screaming Case:
 - Esto se refiere a nombres en mayúsculas.
- Hungarian Notation:
 - Los nombres comienzan con un prefijo en minúsculas para indicar la intención. El resto del nombre está en Pascal Case. Viene en dos variantes:
 - (a) Sistemas húngaros, donde el prefijo indica el tipo de datos;
 - (b) Aplicaciones húngaro, donde el prefijo indica un propósito lógico.



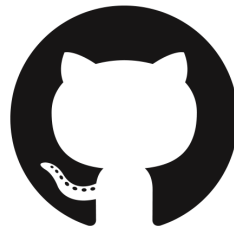
```
1 //Camel Case
2 UserAccount, FedEx, WordPerfect.
3 //Microsoft usa el término Camel Case para referirse estrictamente a esta variación.
4 iPad, eBay, fileName, userAccount.
5
6 //Pascal Case
7 UserAccount. //is in Pascal Case but not userAccount.
8
9 //Snake Case
10 first_name, error_message, account_balance.
11
12 //Kebab Case
13 first-name, main-section.
14
15 //Screaming Case
16 TAXRATE, TAX_RATE.
17
18 //Hungarian Notation
19 // (a) Systems Hungarian, where prefix indicates data type;
20 strFirstName, arrUserNames.
21 // (b) Apps Hungarian, where prefix indicates logical purpose.
22 rwPosition, pchName.
```



Google C++ naming conventions



Nuestro repo de GitHub



Map of Computer Science



¿Cómo es un DÍA de un INGENIERO de SOFTWARE?





- L. J. Aguilar, Programación en C++. Algoritmos, estructuras de datos y objetos, Aravaca (Madrid): McGRAW-HILL, 2006.
- D. Malik, C++ Programming: From Problem Analysis to Program Design, Boston, MA: Cengage Learning, 2003.