



# < SOBRECARGA DE OPERADORES >



< POR OSCAR ALEJANDRO PENILLA SKAKIEVICH >

<4C1>  
<20300701>

<FECHA:23/05/22>

# 1.- DESCRIPCIÓN, CARACTERÍSTICAS Y TIPO

- Es un operador lógico de comparación, se usa para comparar dos objetos de cualquier clase y así entregar valores binarios como resultado.
- Se va a usar cuanto tengamos que buscar una igualdad o verificar si un objeto tenga los mismos valores que otro.

# VENTAJAS Y DESVENTAJAS

==

- Ventajas
  - Ayuda a comparar dos objetos
  - No modifica los valores involucrados
  - Envía una respuesta cerrada
  - Acepta datos de todo tipo para comparar
- Desventajas
  - Tiene usos muy cerrados y puntuales

- (unario)

- Ventajas
  - Ayuda a hacer negativo un objeto numeric
  - Puede usarse en diferentes procesos automáticos como en formulas donde se necesite un valor negado pero también positivo.
- Desventajas
  - Se puede confundir con el signo binario sustracción “-”
  - Cambia o crea un nuevo objeto al negarlo
  - Solo acepta datos numericos

# SINTAXIS DENTRO Y FUERA

- Dentro de la clase

- `<tipo> operator==( <variables> )`

- Fuera de la clase

- `<nombre de la clase> operator==( <variables> ){`  
    `<Sentencias>`  
    `};`

# EJEMPLO DE CODIGO

```
#include <iostream>

using namespace std;

class time
{
    int hr, min, sec;
public:
    // default constructor
    time()
    {
        hr = 0, min = 0; sec = 0;
    }

    // overloaded constructor
    time(int h, int m, int s)
    {
        hr = h, min = m; sec = s;
    }

    //overloading '==' operator
    friend bool operator==(time& t1, time& t2);
};
```

```
bool operator==(time& t1, time& t2)
{
    return (t1.hr == t2.hr && t1.min == t2.min &&
t1.sec == t2.sec);
}

void main()
{
    time t1(3, 15, 45);
    time t2(4, 15, 45);
    if (t1 == t2)
    {
        cout << "Ambos valores son iguales";
    }
    else
    {
        cout << "Ambos valores son diferentes";
    }
}
```

# PANTALLA DE COMPILACIÓN

```
Ambos valores son diferentes
C:\Users\penil\OneDrive\Escritorio\CETI\PROGRAMACION\
Para cerrar automáticamente la consola cuando se detie
puración.
Presione cualquier tecla para cerrar esta ventana. . .
```

# BIBLIOGRAFÍAS

- *C++ => Sobrecarga del operador.* (s. f.). learntutorials. <https://learntutorials.net/es/cplusplus/topic/562/sobrecarga-del-operador#:~:text=Puede%20sobrecargar%20todos%20los%20operadores,%3E%3D%20y%20%3C%3D>
- *C++ operator== overloading.* (2012, 9 abril). Stack Overflow. <https://stackoverflow.com/questions/10070020/c-operator-overloading>
- *C++ Operator Overloading Examples | Studytonight.* (s. f.). Studytonight. <https://www.studytonight.com/cpp/operator-overloading-examples>
- C. (2021, 3 agosto). *Equality operators: == and !=.* Microsoft Docs. <https://docs.microsoft.com/en-us/cpp/cpp/equality-operators-equal-equal-and-exclpt-equal?view=msvc-170>
- C. (2022, 2 mayo). *Operadores binarios.* Microsoft Docs. <https://docs.microsoft.com/es-es/cpp/cpp/binary-operators?view=msvc-170>
- *C Con Clase / Curso C++ (cap35).* (s. f.). conclase. <https://conclase.net/c/curso/cap35>
- Gautam, A. (2021, 29 abril). *How to Overload Operators in C++.* freeCodeCamp.Org. <https://www.google.com/amp/s/www.freecodecamp.org/news/how-to-overload-operators-in-cplusplus/amp/>
- GeeksforGeeks. (2021, 18 mayo). *C++ program to compare two Strings using Operator Overloading.* <https://www.geeksforgeeks.org/c-program-to-compare-two-strings-using-operator-overloading>