**OPERACIONES**

**CADA APARTADO PRINCIPAL CORRESPONDE CON EL NOMBRE DEL PROYECTO CON EJEMPLOS DE LO APUNTADO AQUÍ**

1. **OPERACIONES BÁSICAS**

* Sumar
* Restar
* Multiplicar
* Dividir
* Calcular el módulo

1. **PRECEDENCIA Y ASOCIATIVIDAD**

* Son el conjunto de reglas que usamos para determinar el orden a la hora de operar cuando hay varios operadores diferentes.
* Precedencia: Que operación hacer primero.
* Asociatividad: En que dirección se hace la operación.



1. **OPERADORES DE INCREMENTO Y DECREMENTO PREFIJOS Y POSTFIJOS**

* Decir “incrementar una variable” significa aumentar el valor de la misma.
* Decir “decrementar una variable” significa disminuir el valor de la misma.
* Podemos usar ++ como prefijo o postfijo de la variable para aumentar su valor en 1.
* Podemos usar -- como prefijo o postfijo de una variable para disminuir su valor en 1.
* La diferencia entre el prefijo y el postfijo, es cuando se realiza el aumento o decremento de la variable, si es un prefijo se realizará antes de ejecutar toda la sentencia y si es un postfijo se realizará tras la ejecución de la sentencia.
* Hay ejemplos más explicativos en los proyectos.
* Estos prefijos y postfijos solo sirven para aumentar o para disminuir en 1 el valor de la variable. No existen estos operadores para multiplicar o dividir.

1. **OPERADORES COMPUESTOS**

* Permiten hacer operaciones de una forma más simplificada a la hora de escribir, pero su uso principal es el de asignar el nuevo valor de la variable directamente tras realizar la operación.
* Para usar operadores compuestos, ponemos el símbolo de la operación que queremos realizar (suma, resta, multiplicación, división y módulo) seguido de un igual y el valor por el que queremos operar con nuestra variable. Ejemplo:

int x = 5;

x += 6; //Es lo mismo que poner x = x + 6

std::cout << “x: ” << x << std::endl; // x = 11

1. **OPERADORES RELACIONALES**