

Actividad | 2# | Operaciones Básicas

en C++

Lenguajes de Programación I

Ingeniería en Desarrollo de Software



TUTOR: Francisco Zapien Bueno

ALUMNO: Casandra Montserrat ortiz Cortes G-1

FECHA: 30/03/2024

Índice

Introducción.....1

Descripción.....2

Justificación.....3

Desarrollo.....4

- Código
- Prueba del sistema

Conclusión.....5

Referencia

INTRODUCCIÓN

C++ proporciona un conjunto muy rico de operadores y define cuáles de estos operadores se aplican a tipos fundamentales. Una expresión se compone de uno o más operandos que se combinan mediante operadores. La forma más simple de una expresión consta de una sola constante literal o variable. Las expresiones más complejas se forman con un operador y uno o más operandos. Cada expresión produce un resultado. Existen operadores unitarios que actúan sobre un solo operando (operador dirección &; operador de referencia *) y operadores binarios y terciarios, que actúan sobre dos o tres operandos (suma +, resta -). Los operadores se clasifican en función de las tareas que ejecutan: aritméticos, relacionales, lógicos.

Operadores aritméticos.		
Operador	Función	Ejemplo
+	más unitario	+ expr
-	menos unitario	- expr
*	multiplicación	a * b
/	división	a / b
%	resto (módulo)	a % b
+	suma	a + b
-	resta	a - b

Operadores relacionales y lógicos.		
Operador	Función	Ejemplo
!	NOT lógico	!expr
<	menor que	expr1 < expr2
<=	menor o igual que	expr1 <= expr2
>	mayor que	expr1 > expr2
>=	mayor o igual que	expr1 >= expr2
==	igualdad	expr1 == expr2
!=	desigualdad	expr1 != expr2
&&	AND lógico	expr1 && expr2
	OR lógico	expr1 expr2

Operador lógico OR ().		
	expr1 == true	expr1 == false
expr2 == true	true	true
expr2 == false	true	false

Operador lógico AND (&&).		
	expr1 == true	expr1 == false
expr2 == true	true	false
expr2 == false	false	false

Operadores de asignación.	
Operador	Significado
<code>=</code> <code>*=</code> <code>/=</code> <code>%=</code> <code>+=</code> <code>-=</code> <code><<=</code> <code>>>=</code> <code>&=</code> <code>^=</code>	Asignación simple <code>m operador = n ;</code> equivale a <code>m = m operador n</code>
Operadores de incremento y decremento.	
Operador	Significado
<code>++</code>	Incremento posfijo (<code>a++</code>)
<code>--</code>	Decremento posfijo (<code>a--</code>)
<code>++</code>	Incremento prefijo (<code>++a</code>)
<code>--</code>	Decremento prefijo (<code>--a</code>)

Las sentencias de control de un lenguaje de programación determinan el flujo de ejecución a través de un programa. Existen tres categorías: Secuencia, Selección y Repetición. Una sentencia compuesta es un grupo de sentencias que se ejecutan.

DESCRIPCIÓN

En las operaciones básicas nos permite realizar el dominio de la suma , resta, multiplicación o división; aplicando en los números operaciones fundamentales que nos ayuda en cualquiera realización como, por ejemplo: en empresas, en la tecnología , trabajos escolares, incluyendo en el hogar o fuera.

Lo que es fundamental en nuestra vida diaria , aplicando los números en cualquier momento, se nos inculca las operaciones básicas desde que vamos al jardín de niños aprender como se hace alguna suma o resta de manera que podamos resolverla y entendamos y podamos memorizar en nuestro ciclo de vida hasta llegar a la adultez.

Lo que nos damos cuenta que las operaciones van evolucionando al paso del tiempo lo que cada vez vamos aprendiendo diferentes maneras o formas de realizarse y son nuevos retos para nosotros ya que las matemáticas podemos verlos en todos lados y en cualquiera momento en alguna actividad.

JUSTIFICACIÓN

El valor devuelto por main() para determinar si el programa ha tenido o no éxito. Un valor de retorno 0 indica éxito. La función main() debe existir en cada programa C++ y es la única función que llama explícitamente el sistema operativo. Un programa C++ completo que visualice la frase en español "Hola mundo" es

```
// programa hola.cpp
// el programa visualiza "Hola mundo"
#include <iostream> // declaraciones
using namespace std;
int main() // función main( )
{
// visualizar "Hola mundo"
cout<< "Hola mundo!" << endl;
return 0;
}
```

La directiva using namespace std define el espacio de nombres std y evita tener que utilizar esta directiva. Si no se utiliza el espacio de nombres using namespace std la salida con cout requerirá ser precedida por el espacio de nombre std.

```
#include <iostream> int main ()
{
// visualizar "Hola mundo"
std::cout<< "Hola mundo" << std::endl;
return 0;
}
```

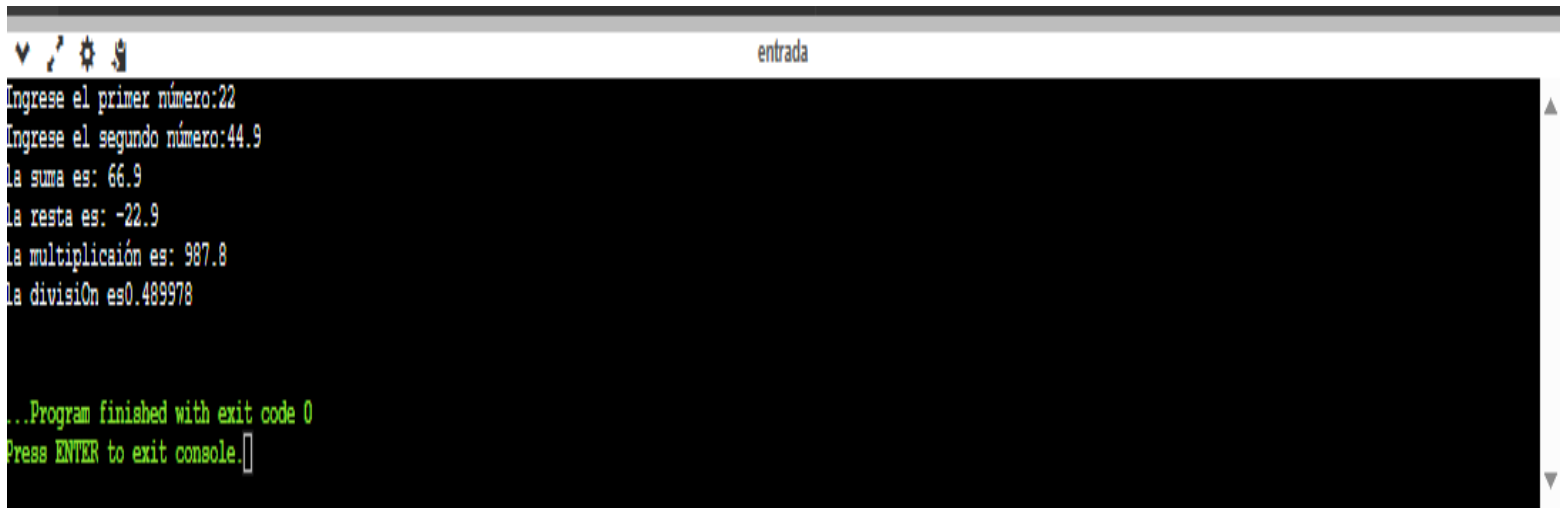
DESARROLLO

1. Que pida el ingreso de dos valores (ya sean enteros o decimales).
2. Que los sume, reste, multiplique y divida.
3. Que los muestre en pantalla con su respectivo enunciado.

Para ello, se debe tener en cuenta la siguiente estructura de impresión de datos:

Escribe el primer número:
Escribe el segundo número:
La suma es:
La resta es:
La multiplicación es:
La división es:

- Código

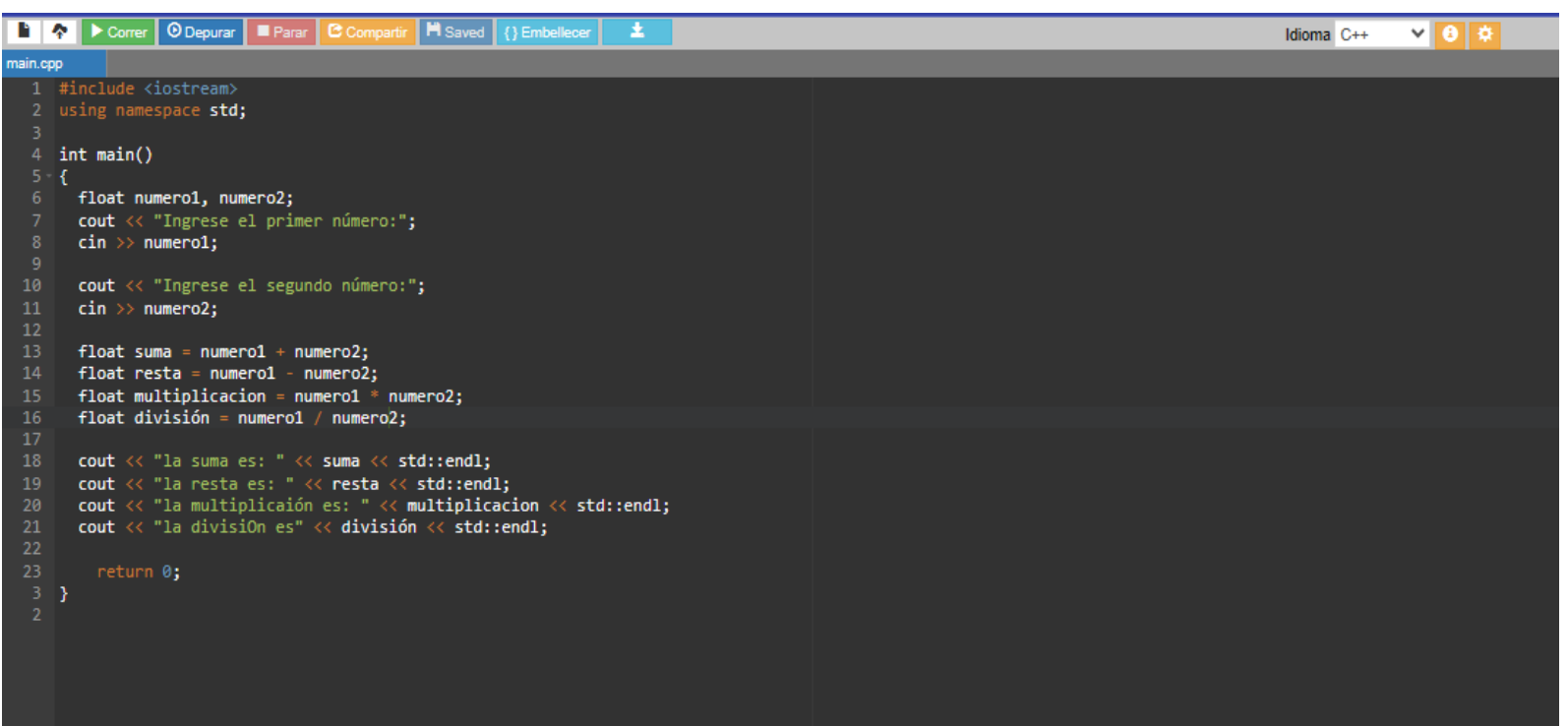


A screenshot of a terminal window titled 'entrada'. The terminal shows the execution of a C++ program. It prompts for two numbers: 'Ingrese el primer número:22' and 'Ingrese el segundo número:44.9'. It then displays the results of calculations: 'La suma es: 66.9', 'La resta es: -22.9', 'La multiplicación es: 987.8', and 'La división es: 0.489978'. The program ends with the message '...Program finished with exit code 0' and 'Press ENTER to exit console.'.

```
entrada
Ingrese el primer número:22
Ingrese el segundo número:44.9
La suma es: 66.9
La resta es: -22.9
La multiplicación es: 987.8
La división es: 0.489978

...Program finished with exit code 0
Press ENTER to exit console.
```

- Prueba del sistema



A screenshot of a C++ IDE showing the source code of the program. The code is in a file named 'main.cpp'. It includes the necessary headers and uses the std namespace. The main function prompts for two numbers, calculates the sum, difference, product, and quotient, and prints the results. The code is as follows:

```
1 #include <iostream>
2 using namespace std;
3
4 int main()
5 {
6     float numero1, numero2;
7     cout << "Ingrese el primer número:";
8     cin >> numero1;
9
10    cout << "Ingrese el segundo número:";
11    cin >> numero2;
12
13    float suma = numero1 + numero2;
14    float resta = numero1 - numero2;
15    float multiplicacion = numero1 * numero2;
16    float división = numero1 / numero2;
17
18    cout << "la suma es: " << suma << std::endl;
19    cout << "la resta es: " << resta << std::endl;
20    cout << "la multiplicación es: " << multiplicacion << std::endl;
21    cout << "la división es" << división << std::endl;
22
23    return 0;
24 }
```

CONCLUSIÓN

Las aplicaciones prácticas en la vida cotidiana son aquellas herramientas digitales que nos facilitan tareas y actividades diarias. Ya sea a través de aplicaciones móviles o plataformas web, estas herramientas tienen un impacto significativo en la forma en que llevamos a cabo diversas actividades.

En la actualidad, las aplicaciones prácticas abarcan una amplia gama de funciones, desde la gestión del tiempo y la productividad, hasta la salud, el fitness y la educación. Estas herramientas nos permiten optimizar nuestro tiempo y recursos, mejorar nuestra salud y bienestar, y acceder a información relevante de manera rápida y eficiente.

Desde aplicaciones que ayudan a planificar y organizar tareas diarias, hasta aquellas que fomentan la meditación y el cuidado personal, las herramientas digitales están cada vez más integradas en nuestras rutinas diarias, brindando soluciones prácticas para mejorar nuestra calidad de vida, consecuencias de no dominar las operaciones básicas en cualquier campo pueden ser significativas.

Referencias

¿Por qué es importante aprender las operaciones básicas? (s.f.). Obtenido de <https://apolonio.es/por-que-es-importante-aprender-las-operaciones-basicas/>

Apéndice II Guía de sintaxis del lenguaje ANSI/ISO C++ estándar. (2014). Obtenido de https://highereducation.com/sites/default/files/6071514681/1131128/Guia_de_sintaxis_de_C_plus_plus.pdf

[https://github.com/casandraortiz31/Casandra-](https://github.com/casandraortiz31/Casandra-Ortiz/blob/efef7d8adffefcf077fc6b0a331b08b51bdb72c3/main%20(1).cpp)

[Ortiz/blob/efef7d8adffefcf077fc6b0a331b08b51bdb72c3/main%20\(1\).cpp](https://github.com/casandraortiz31/Casandra-Ortiz/blob/efef7d8adffefcf077fc6b0a331b08b51bdb72c3/main%20(1).cpp)