

Actividad | #2| Operaciones básicas en C++

Lenguajes de Programación I

Ingeniería en Desarrollo de Software

TUTOR: Urbano Francisco Ortega Rivera

ALUMNO: Alejandra Ibarra Carmona

FECHA: 28/07/2024

Índice

Introducción ………………………………………………………3

Descripción ……………………………………………………….3

Justificación ………………………………………………………3

Desarrollo …………………………………………………………4

* Código…………………………………………………….4
* Prueba del sistema………………………………………5

Conclusión………………………………………………………7

Referencias………………………………………………………7

Introducción

En esta actividad se estarán presentando y ejecutando un código que nos ayude a realizar operaciones básicas en lenguaje C++. Podemos conocer bastantes metodologías que nos ayuden a construir programas y conlleva etapas las cuales son:

* Análisis: en esta etapa se busca el problema a resolver sin olvidar incluir el contexto en el cual se usará.
* Diseño: aquí se muestra el como se va a resolver nuestro problema planteado, esto se debería de realizar por escrito ya que ahí quedarán todas las decisiones estas a su vez se utilizarán como dato en el contexto de la realidad como una solución
* Implementación: Esto es traducir a un lenguaje de programación el diseño anterior.

Las estructuras de control se sentencias son aquellas que bifurcan la ejecución del programa, en cuanto se llega a este punto se puede ejecutar un grupo de instrucciones u otros o bien repetir el ciclo un determinado número de veces. Así este tipo de estructuras nos permiten la repetición de un grupo de instrucciones mientras se cumpla la condición.

Descripción

En este lenguaje que utilizamos para esta actividad que es el C++ orientado a objetos, existen tres principales estructuras de control:

* While: En este caso la sentencia se ejecutará repetidamente siempre y cuando la condición sea verdadera, si esta es falsa no se ejecutará ninguna vez.
* Do While: En este caso a las sentencias se les permite ejecutar al menos una vez sin tener esa condición como el anterior, esto a su vez que se ejecutan se evalúan, se repiten las sentencias hasta que la condición sea falsa o mientras un grupo de sentencias se ejecute y sea verdadero esto nos demuestra que por lo menos una vez tienen que ser aplicadas y ejecutadas.
* For: En este caso se repite la sentencia cuando ya se conoce el numero de veces a realizarlo, para que se pueda cumplir primero necesitamos la inicialización, no dejemos de lado la variable que servirá para controlar el numero de veces que el bucle se repita, aunque se mencione al último también necesitamos una condición y un incremento esto para modificar el valor de la variable de control.

Justificación

A diferencia de las tres principales que se mencionaron, existen otras orientadas a los bucles o ciclos, y estas son selectivas y se enfocan en bifurcaciones, se utilizan para establecer una posible ruta con un grupo o bloque de instrucciones y se mencionan los siguientes tipos:

* If: Esta se basa en el resultado ya sea verdadero o falso en una expresión, para esto se necesita una condición y una sentencia en la primera se ejecutará y se evalúa como verdadera o falsa y la segunda es la que determina si es verdadera.
* If else: En esta se realizan acciones diferentes cuando la condición sea verdadera y así como también cuando es falsa.
* Switch: Es una instrucción de decisión múltiple, en ella se comparan el valor de una expresión con lista de constantes de carácter tipo entero, en caso de que el valor de la expresión corresponda con alguna de las constantes se ejecutan las acciones asociadas a esta.

Desarrollo

* Código

#include <iostream>

int main () {

float numero1, numero2;

std::cout <<"Escribe el primer numero: ";

std::cin >> numero1;

std::cout <<"Escribe el segundo numero: ";

std::cin >> numero2;

float suma = numero1 + numero2;

float resta = numero1 - numero2;

float multiplicacion = numero1 \* numero2;

float division = numero1 / numero2;

std::cout <<"La suma es: " << suma << std::endl;

std::cout <<"La resta es: " << resta << std::endl;

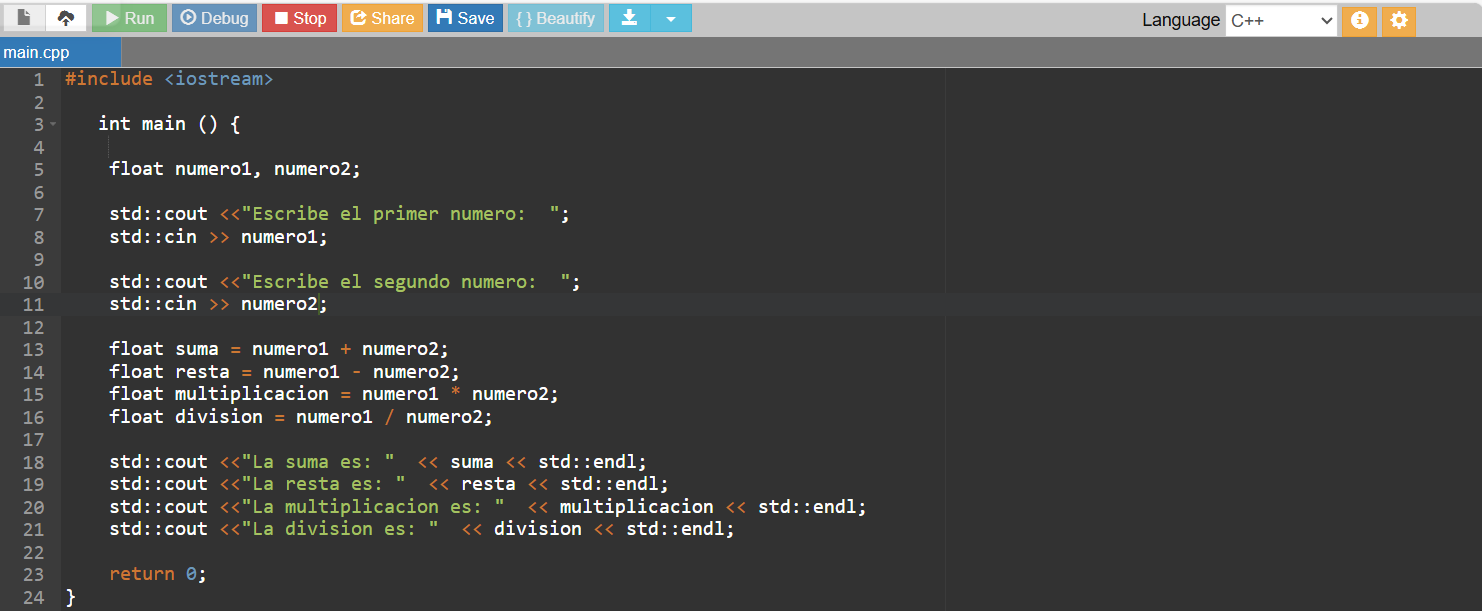
std::cout <<"La multiplicacion es: " << multiplicacion << std::endl;

std::cout <<"La division es: " << division << std::endl;

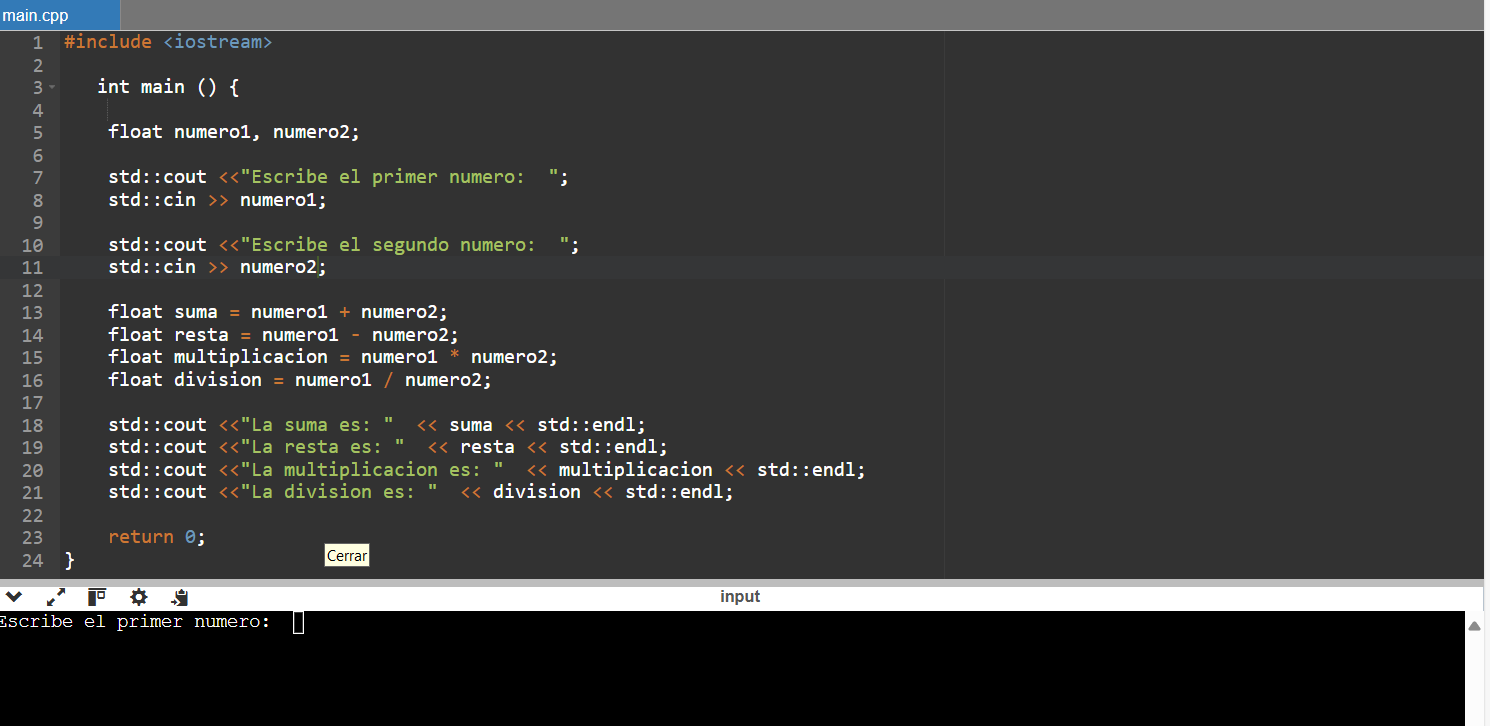
return 0;

}

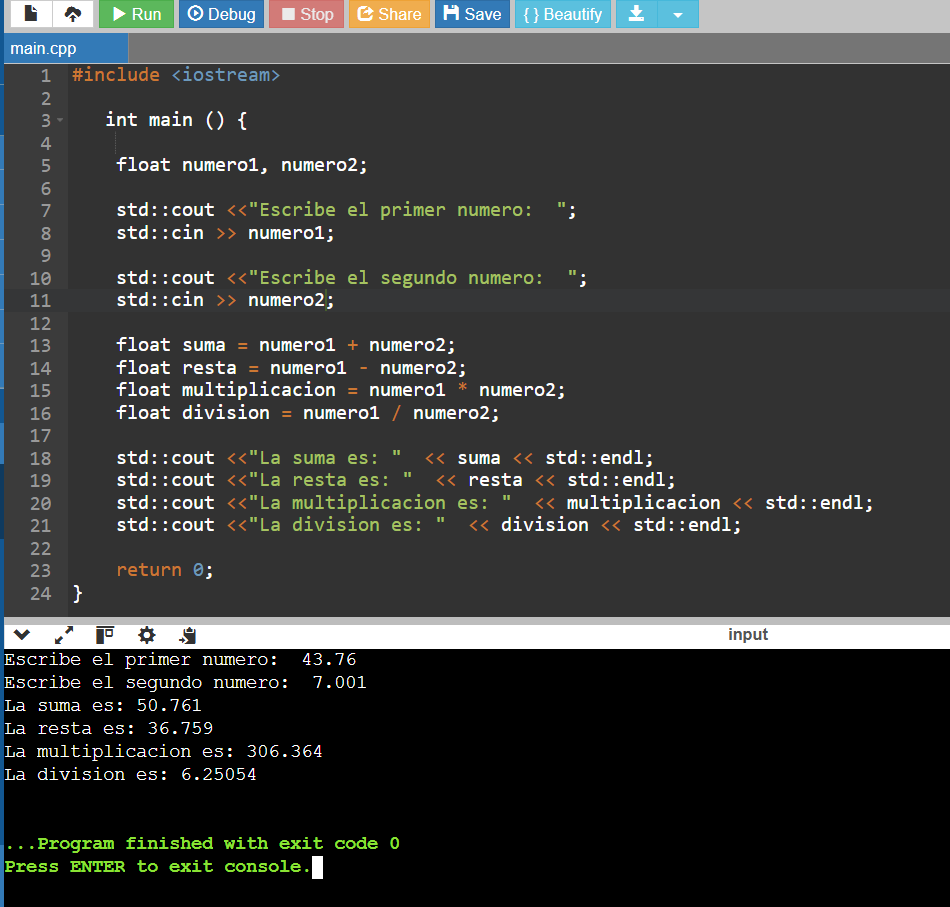
* Pruebas del sistema
* En esta imagen se aprecia el código escrito para la realización de las operaciones aritméticas en grupo, es decir al ingresar los dos valores nos dará los cuatro resultados seguidos



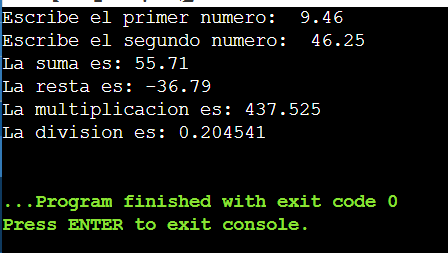
* Aquí ya se muestra el codigo ejecutado listo para ingresar los dos numero



* En esta imagen ya se ejecuta el programa de forma correcta con los 2 valores ingresados



* Otro ejemplo de la funcionalidad con diferentes valores



Conclusión

Llegamos a la conclusión de que en esta actividad el código descrito nos muestra las operaciones básicas, pero en la actividad nos especifica que se ingresen dos números y que se apliquen en forma de grupo las operaciones. Cuando se esta desarrollando cualquier tipo de programa informático comúnmente se utilizan diferentes operadores para llevar a cabo los procesos aritméticos en este caso los que nos está pidiendo la actividad.

Mencionando a los operadores en este lenguaje son pieza fundamental ya que nos proporcionan diferentes operaciones a realizar las cuales utilizan el tipo de suministro necesario para ser ejecutadas. En el lenguaje C++ lo utilizamos para lo básico, realizar innumerables tareas y resolver cualquier tipo de estructuras, en nuestra vida cotidiana el proceso de resolución de un problema de una computadora nos orilla a realizar una escritura de un programa y su posterior ejecución, recordemos que también son importantes las sentencias y las estructuras de nuestro programa.

Referencias

(s.f.). Obtenido de GitHub: <https://github.com/Ale182401/Lenguaje-de-Programaci-n-1/tree/7135649f08f14f2fe53e45175cbfe4e1cf240ee7>