
Semana # 2– Utilizando C++, estructuras, condiciones, operadores y operandos

INTRODUCCIÓN / ESTRUCTURAS

El lenguaje C facilita un método estructurado y disciplinado para el diseño de programas de computación.

Esta semana vamos a aprender los elementos básicos de un programa en C++, cómo mostrar un mensaje en pantalla, cómo capturar una variable, declararlas, operadores matemáticos, etc.

Pero antes, ¿qué es una instrucción?

Una instrucción es lo que le ordenamos a la máquina para que ejecute, por eso se dice que un programa es un conjunto de instrucciones; ya que cuando ejecutamos un programa, se ejecutan así mismo en su interior muchas instrucciones que hacen que la maquina realice algo.

Instrucciones declarativas –

Sirven para declarar librerías, variables, constantes, arreglos, punteros, estructuras...

Por ejemplo:

1.) Para librerías:

Las librerías las declaramos porque vamos a usar recursos que contienen ellas

`#include <iostream.h> —> cin , cout..`

`#include <conio.h> —> funcion getch()..`

`#include <string.h> —> para manipular cadenas`

`#include <math.h> —> para operaciones numericas`

`#include <time.h> —> para usar la hora`

`#include <stdio.h> —> para usar alternativas de entrada – salida como printf , scanf`

`#include <stdlib.h> --- permite un comando externo- system("pause"),system("cls")`

2.) Para variables:

A toda variable que se use en un programa, se le debe asociar (generalmente al principio del programa) un tipo de dato específico.

Un tipo de dato define todo el posible rango de valores que una variable puede tomar

al momento de ejecución del programa y a lo largo de toda la vida útil del propio programa.

Los tipos de datos más comunes en C++ son:

Tipos de datos	Longitud del dato
long double	80 bits 3.4×10^{-4932} a $1.1 \times 10^{+4932}$
double	64 bits 1.7×10^{-308} a $1.7 \times 10^{+308}$ (15 dec)
float	32 bits 3.4×10^{-38} a $3.4 \times 10^{+38}$ (6 dec)
unsigned long int	32 bits 0 a 4,294,967,295
long int	-2,147,483,648 a 2,147,483,647
unsigned int	32 bits 0 a 4,294,967,295
int	32 bits -2,147,483,648 a 2,147,483,647
short	16 bits -32,768 a 32,767
char	8 bits -128 a 127

Consejo: A las únicas variables que se deben dar valores iniciales son a:

- los contadores
- los acumuladores

Pero ¿Cuál es la diferencia entre ambos?

Acumuladores: se incrementa o decrementa en un valor variable.

Ejemplo: sacar el promedio de un alumno, se suman las notas (que varían) y se divide para el número de notas.

Contadores: se incrementa o decrementa en una forma constante. Ejemplo: va contando de “1 en 1” ó de “-3 en -3”, etc...

Operadores aritméticos en C++

Operación en C	Operador aritmético
Suma	+
Resta	-
Multiplicación	*
División	/

Operadores de Relación

Se utilizan para expresar condiciones y describen una relación entre 2 variables. El conjunto de operaciones relacionales se muestra en la siguiente tabla.

Operadores		Significado
<		Menor que
>		Mayor que
==		Igual a
>=		Mayor o igual que
<=		Menor o igual que
!=		Distinto o diferente a

Orden de Evaluación

Al incluir estos operadores con los operadores aritméticos, obtenemos este nuevo orden de evaluación:

Prioridad		Operador
1		Contenido de parentesis ()
2		*, /, div, mod
3		+, -
4		==, <, >, <=, >=, !=

NOTA:

En la librería <stdio.h> tenemos para la entrada y salida de datos a scanf y printf respectivamente.

Uso:

– Para la entrada de datos (pedir)
scanf (" %i %s %f " , &edad apellido &sueldo) ;

Nota: Los ampersand (&) van en todo tipo de variable menos en las cadenas de caracteres cuando usamos el scanf.

– Para la salida de datos (mostrar)
printf (" su edad es: %i " , edad)

[*] Sin Nombre1

```
1 //Nuestro primer programa
2 #include<iostream>
3 #include<conio.h>
4 /*Librerias:permiten al compilador cargar o interpretar
5 algunas palabras reservadas para la herramienta*/
6
7 using namespace std; /*Permite leer el lenguaje bajo la biblioteca
8 | | | | | iostream*/
9
10 main() /*Sentencia donde se empieza a ejecutar el programa, define
11 | | | | | el el cuerpo del programa*/
12 { /*Llave izq: inicia el cuerpo de la funcion, y se utiliza en ciclos,
13 | | | | | sentencias if, entre otros*/
14
15 //DECLARACION DE VARIABLES
16 int num1,num2,resultado; /*Declaramos tres variables enteras*/
17
18 cout<<"Digite el primer numero\n "; /*Asi se muestra un comentario
19 | | | | | en pantalla*/
20 /* el \n y el endl es un salto de linea*/
21 cin>>num1; /*Asi se captura o guarda un valor en tiempo de ejecucion*/
22 cout<<"Digite el segundo numero\n ";
23 cin>>num2;
24 //APLICACION DE LA OPERACION//
25 resultado=num1+num2; /*Le asigna el resultado de la suma a la variable
26 | | | | | resultado*/
27 cout<<"El resultado de la suma es: "<<resultado<<endl;
28 /*Asi mostramos el resultado en pantalla*/
29
30 getch(); /*Mantiene pausado el programa; con el fin de verificar que el
31 | | | | | programa ejecuta correctamente, con libreria conio.h*/
32 return 0; /*Mantiene el ultimo mensaje en pantalla, en el momento que el
33 | | | | | usuario digite ENTER el programa sale o continua en alguna seccion del
34 | | | | | codigo segun sea el caso*/
35 } /*Llave der: cierra el cuerpo del programa.
```