

Programa

1. Componentes Básicos de Un

Componentes

Un programa en C++ está definido por funciones (grupo de instrucciones que pueden o no hacer algún cálculo), donde la función

principal debe ser llamada **main**. La composición general de un programa en C++ es:

- directivas de preprocesamiento.
- declaración globales.
- funciones definidas por el usuario
- función main
- comentarios para entender el funcionamiento del programa

Directivas

Las directivas de preprocesamiento son definidas para que el compilador realice algo antes de compilar el programa (revisar si la

sintaxis es correcta y generar un código ejecutable por la computadora), como el incluir funciones de alguna biblioteca conocida

en particular.

En C la directiva para incluir bibliotecas es como las siguientes:

Ejemplos:

```
#include <iostream.h>
```

Esta directiva permitirá utilizar la función de lectura **cin** y la función de escritura **cout** propias de C++.

```
#include <stdio.h>
```

Esta directiva permitirá utilizar las funciones de lectura y escritura propias de C.

Estas directivas también funcionan con C++, aunque existen algunos compiladores que no las

soportan en cuyo caso es necesario

hacer uso de los namespaces.

Un namespace es una declaración de región. El propósito de estos es localizar el nombre de identificadores para evitar colisiones.

Para tener compatibilidad, cuando un programa en C++ incluye una librería de C, haremos uso del namespace.

Ejemplo:

En C tenemos `#include <iostream.h>`

con el respectivo en C++ `#include <iostream>`
`using namespace std;`

Declaraciones Globales

Las declaraciones globales son definiciones de variables o constantes que serán utilizadas por cualquiera de todas las funciones definidas en el programa.

La sintaxis para la declaración de una **variable** es:

Tipo var1, var2,..., var3;

o bien

Tipo var1 = valor, var2,..., var3;

Ejemplos:

int edad;

Esta declaración indica que la variable edad definida como un valor entero puede ser utilizada en cualquier función.

double venta = 0.0;

Esta declaración presenta la declaración de una variable como valor doble a la cual se le da un valor inicial y puede ser usada en cualesquier función.

char nombre[30];

Esta declaración es para declarar una variable de tipo cadena de caracteres.

La sintaxis para la declaración de una **constante** es:

const double PI = 3.1415;

Esta declaración representa el valor de la constante PI la cual podrá utilizarse en cualquier

función.

Reglas para nombre de variables

- Deben empezar con letra o underscore
- Pueden contener letras, números y underscore
- Las letra mayúsculas y minúsculas son diferentes para C++

Funciones definidas por el usuario y función main

La definición de una función se realiza escribiendo primero el tipo del valor de retorno de la función, después el nombre de la función, posteriormente entre paréntesis las variables que utilizará dicha función (parámetros) y finalmente las instrucciones de la función. Ejemplos:

```
double promedio( int a, int b, int c)
```

```
{  
  return (a + b + c) / 3.0;  
}
```

Declara a la función promedio, la cual recibe tres valores enteros y calcula y regresa el promedio de ellos.

```
int main ()
```

```
{  
  cout << " Función main desplegando un mensaje ";  
  return 0;  
}
```

Declara a la función principal main, no tendrá parámetros, ni regresará algún valor.

A través de las funciones cin y cout se pueden tomar datos del teclado y desplegar información correspondientemente en la computadora, dichas funciones pertenecen a la biblioteca iostream.h

Comentarios

Los comentarios pueden escribirse utilizando alguna de las siguientes formas:

1. comentando en la misma línea, utiliza //, ejemplo

```
int edad; // la edad se utilizará como un valor entero.
```

2. comentando entre varias líneas, utiliza /* */ , ejemplo:

```
/* La siguiente función promedio recibe tres valores enteros
   y calcula y devuelve el promedio de los tres a través
   de un valor real */
```

NOTA IMPORTANTE

Todos los ejemplos de este curso están basados en el compilador Borland TurboC++ sugerido.

Ejemplos de Programas

A continuación tendrás algunos ejemplos de programas en C++:

Ejemplo I

```
#include <iostream.h> // biblioteca para el manejo de lectura y escritura

int edad;
int main()
{
    cout << "Da la Edad "; // se despliega el
mensaje a la pantalla
    cin >> edad; // se lee el dato del
teclado (la edad)
    if (edad >= 18) // validando si la edad
es mayor o igual a 18
        cout << "Eres mayor de Edad"; // "Mayor de Edad" es
desplegado
    else
        cout << "Eres menor de Edad"; // "Menor de Edad" es
desplegado
    return 0;
}
```

Ejemplo II

```
#include <iostream.h>

/* Declaración de variables globales y funciones.
En este ejemplo no hay variables globales, solo
declaramos la función que calculará el promedio de
3 calificaciones que son leídas en la función main
y enviadas a la función promedio */

/*
Funcion promedio que recibe tres datos y
regresa el cálculo del promedio de los tres */

double promedio(int a, int b, int c)
{
    return (a + b + c) / 3.0;
}

int main()
{
    int edad1, edad2, edad3; // declaramos las variables
    locales
    como
    cout << "Da la Edad 1";
    cin >> edad1;
    cout << "Da la Edad 2";
    cin >> edad2;
    cout << "Da la Edad 3";
    cin >> edad3;
    cout << "El promedio de las 3 edades es = " << promedio (edad1,
    edad2, edad3);
    return 0; /* se envían como
    parámetro
}

</SPAN< p>
```

Ejercicio

Baja el compilador de Borland turboC++ en la dirección:

http://www.borland.com/products/downloads/download_cbuilder.html#

Baja el editor de Crimson en la siguiente dirección:

<http://www.crimsoneditor.com/english/download.html>

Instala el compilador dando doble clic al archivo de instalación: **freecommandLinetools.exe**

Baja de los recursos los archivos BCC.BAT y RUN.BAT y copialos en la ruta
c:\Borland\BCC55\Bin

Instala el editor Crimson dando doble clic al archivo de instalación **crimsoneditor.exe**

Configura el editor de Crimson usando los siguientes pasos:

- Ejecuta el editor, al hacerlo por primera vez aparece un mensaje de error, solo da Aceptar
- En el Menú **Tools**, selecciona **Preferences**
- En la ventana de Categorías, selecciona **Syntax** y elije **C/C++**
- En la misma ventana, selecciona ahora **User Tools** y has clic donde dice **Empty Ctrl 1** y llena los recuadros de la siguiente forma:

Menú Text : Compilar (puedes poner cualquier texto, es solo el mensaje que aparecerá en la ventana de Tools, que indica la opción de compilar el programa).

Command: `c: \borland\bcc55\bin\bcc.bat`

Argument: haciendo clic en la flecha selecciona *File Name*

Initial dir: haciendo clic en la flecha selecciona *File Dir*

Las opciones **Close on exit**, **Capture output** y **Use shor filename** se dejan sin seleccionar. La opción **Save before execute** queda seleccionada.

Ahora has clic donde dice **Empty Ctrl 2** y llena los recuadros de la siguiente forma:

Menú Text : Ejecutar (puedes poner cualquier texto, es solo el mensaje que aparecerá en la ventana de Tools, que indica la opción de ejecutar el programa).

Command: `c: \borland\bcc55\bin\run.bat`

Argument: haciendo clic en la flecha selecciona *File Title*

Initial dir: haciendo clic en la flecha selecciona *File Dir*

Las opciones **Close on exit**, **Capture output** y **Use shor filename** se dejan sin seleccionar. La opción **Save before execute** queda seleccionada.

Por último da Apply.

Prueba cuidadosamente los ejemplos, revisando que sucede con diferentes valores.

Ligas sugeridas

<http://www.cplusplus.com/doc/tutorial/>

<http://www.cs.wustl.edu/~schmidt/C++/>

[Regresar](#)

[Siguiente módulo](#)