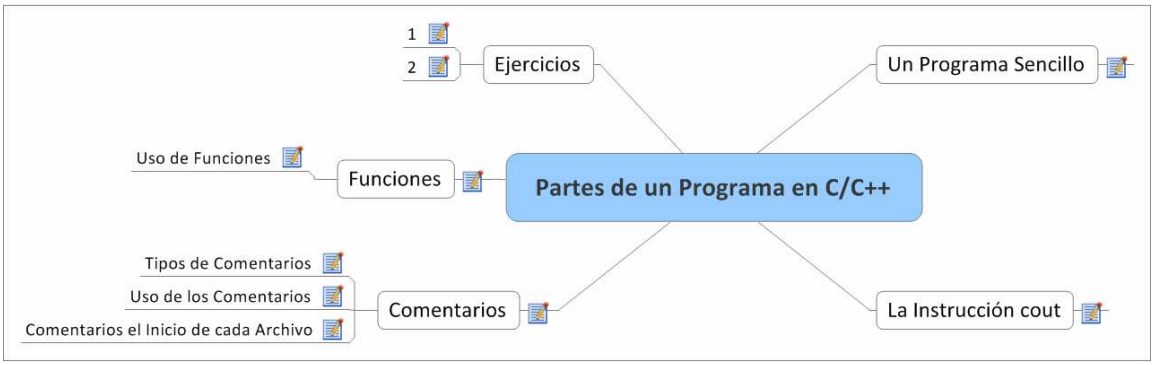
PARTES DE UN PROGRAMA EN C/C++



Un Programa Sencillo

Considérese el programa transcrito a continuación:

```
1:  #include <iostream.h>
2:  void main ()
3:  {
4:        cout << "¡Hola Mundo!\n";
5:  }
```

Aún en un programa tan sencillo como el anterior, que simplemente muestra un mensaje por pantalla, en este caso ¡Hola Mundo!, tiene muchas partes interesantes.

En la línea 1, se incluye el archivo iostream.h. El primer caracter es el símbolo #, el cual indica que lo que está a continuación es una instrucción dirigida al preprocesado. Cada vez que se inicia el compilador, se ejecuta el procesador. El preprocesador lee el código fuente, buscando líneas que comiencen con el símbolo #, y actúa sobre ellas.

La instrucción *include* es una directiva del preprocesador que indica que lo que viene a continuación es el nombre de un archivo cuyo contenido debe ser insertado a partir de ese punto. Los símbolos menor que y mayor que (<>) alrededor del nombre del archivo, le señala al preprocesador que dicho archivo se encuentra en el directorio preestablecido. El archivo iostream.h es necesario para poder utilizar la función *cout*, que es usada para mostrar mensajes por pantalla.

NOTA. El preprocesador se ejecuta antes de compilador, cada vez que este es invocado. El preprocesador traduce todas las líneas que comienzan con el caracter # a comandos especiales, dejando el archivo de código listo para ser compilado.

La línea 2 inicia el programa con la función llamada *main*() . Cada programa en C o C++ tiene una función *main*(). En general, una función es un bloque de código que ejecuta una o más acciones. Normalmente, las funciones son invocadas desde otras funciones, pero main() es especial. Cuando el programa inicia, main() se ejecuta automáticamente.

La función main(), al igual que todas las funciones, debe establecer que tipo de valor retorna. En este programa, el valor de retorno para la función main() es void, lo que significa que no devolverá ningún valor.

Todas las funciones comienzan con una llave abierta ({) y finalizan con una llave cerrada (}). Las llaves en este caso están en las líneas 3 y 5. Todo lo que está dentro de las llaves, se considera parte de la función. Lo más importante de esta función está en la línea 4. Se emplea el objeto *cout* para mostrar un mensaje en la pantalla.

Para usar el objeto cout, debe escribirse seguido del operador de inserción (<<). Lo que siga al operador de inserción será mostrado en pantalla. Si se desea escribir una cadena de caracteres, asegúrese de que esté encerrado en comillas dobles ("), como se muestra en la línea 4.

NOTA. Se le llama cadena de caracteres a una serie de caracteres 'imprimibles'.

Los dos caracteres finales, \n, le indican a cout que añada un salto de línea luego del texto que la precede.

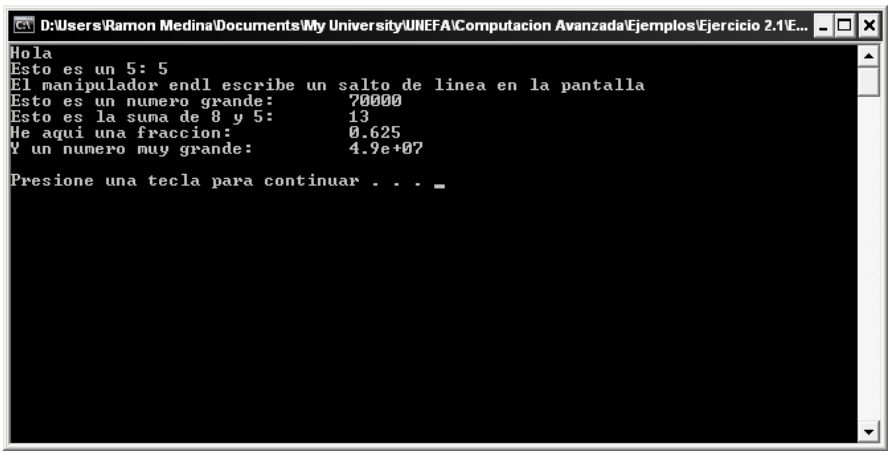
Todos los programas en C/C++ compatible ANSI declaran *main*() retornando una valor de tipo entero (int). Dicho valor es retornado al sistema operativo cuando el programa termina su ejecución. Algunos programadores señalizan un error, devolviendo el valor 1.

La Instrucción *cout*

Para escribir un valor en la pantalla, sólo es necesario usar la palabra cout seguida del operador de inserción (<<), luego de cual se incluye la información que se desea desplegar. Transcriba (sin incluir los números de línea) y ejecute el siguiente programa:

```
1:  #include <iostream.h>
2:  #include <stdlib.h>
3:  int main()
4:  {
5:      cout << "Hola\n";
6:      cout << "Esto es un 5: " << 5 << "\n";
7:      cout << "El manipulador endl escribe un salto de linea en la pantalla" << endl;
8:      cout << "Esto es un numero grande:\t" << 70000 << endl;
9:      cout << "Esto es la suma de 8 y 5:\t" << 8+5 << endl;
10:     cout << "He aqui una fraccion:\t\t" << (float) 5/8 << endl;
11:     cout << "Y un numero muy grande:\t\t" << (double) 7000 * 7000 << endl << endl;
12:
13:     system ("PAUSE");
14:
15:     return 0;
16: }
```

Al ejecutar este programa, se obtendrá un resultado similar al que se muestra en la siguiente figura:



En la línea 1, la sentencia #*include* <iostream.h>, lo cual hace que el contenido del archivo iostream.h sea agregado a su código fuente. Esto es necesario para poder usar el objeto *cout*. La línea 5 muestra el uso más simple de cout, mostrando por pantalla una cadena de caracteres. El símbolo \n es un caracter especial de formato que le dice a cout que añada un salto de línea al final e la cadena.

En la línea 6 se le suministran tres valores a cout, separados entre si por el operador de inserción. El primer valor es la cadena "Esto es un 5: ". Luego un 5 y finalmente un caracter de salto de línea. Esta sentencia genera el texto Esto es un 5: 5 se muestre por pantalla. Como no hay caracter de salto de línea luego de la primera cadena, el siguiente valor es mostrado justo al lado. A esto se le llama concatenación de dos valores.

En la línea 7 se muestra un mensaje informativo, luego de lo cual se emplea el manipulador *endl*. El propósito de *endl* es escribir un salto de línea en la pantalla. En la línea 8 se utiliza un nuevo caracter de formato, \t. Este inserta un caracter de tabulación y es usado en las líneas 8 a la 11, para alinear los resultados. La línea 8 muestra que es posible mostrar enteros largos. En la línea 9 se incluye una suma en la sentencia cout; se le provee el valor 8+5, y en consecuencia muestra como resultado el número 13. En la línea 10 se inserta el valor 5/8. El término (float), le indica a cout que interprete el valor ingresado como un número real (flotante). En la línea 11 se ingresa el valor 7000*7000; el término (double) le dice a cout que interprete el valor como un número real de doble precisión y que lo muestra en notación científica.

Comentarios

Cuando se escribe una programa, siempre es claro y evidente lo que se estuvo haciendo. Sin embargo, al retomar el programa luego de un corto lapso de tiempo durante el cual se estuvo trabajando en algo diferente, para que todo parezca confuso. Para combatir dichas confusiones y ayudar a que otros entiendan el código, se usan los comentarios. Los comentarios son simplemente texto que es ignorado por el compilador, pero que informa al lector acerca de la función específica de una determinada porción de código.

Tipos de Comentarios

Los comentarios en C++ pueden ser de dos tipos:

Comentarios //

Comentarios /*

Los comentarios //, también referidos como comentarios estilo C++, le dicen al compilador que ignore el texto que sigue a // hasta el final de esa línea. Los comentarios /* hacen que el compilador ignore todo hasta que encuentre una marca */. Estas marcas son referidas como comentarios estilo C. Cada /* debe tener su correspondiente */.

Uso de los Comentarios

Como regla general, el programa debe tener comentarios al inicio, indicando lo que éste hace. Cada función debe tener también su comentario, explicando lo que hace y los valores que devuelve. Finalmente, cada sentencia del programa que sea "obscura" o poco obvia, deberá estar comentada. El siguiente programa muestra el uso de comentarios, probando que no afectan el funcionamiento de este.

```
1:  #include <iostream.h>
2:  #include <stdlib.h>
3:  int main()
4:  {
5:      /* Esto es un comentario
6:      que se extiende hasta que
7:      se encuentra una marca estrella-slash */
8:      cout << "Hola!\n";
9:      // este comentario termina al final de esta línea
10:     cout << "El comentario termino!\n";
11:
12:     // los comentarios doble-slash pueden estar solos en una línea
13:     /* al igual que los comentarios slash-estrella */
14:
```

```
15:     system ("PAUSE"); // Esta instrucción detiene la ejecución del programa
16:                       // y espera a que el usuario presione una tecla
17:
18:     return 0;
19: }
```

Las líneas 5 a la 7 son completamente ignoradas por el compilador, así como los comentarios en las líneas 9, 12 y 13. El comentario de la línea 9 termina al final de la línea, mientras que los de las líneas 5 y 13, requieren del caracter */. El comentario de la línea 15 aunque es ignorado por el compilador, no afecta el procesamiento de la sentencia *system*.

Comentarios el Inicio de cada Archivo

Aunque los comentarios incluidos en un programa dependen de cuánta información desea el programador proporcionar, como buena práctica de programación, se aconseja incluir al inicio de cada programa o función, al menos la siguiente información:

Nombre del programa o función

Nombre del archivo

Qué hace el programa o la función

Una descripción de cómo opera el programa o función

El nombre del autor

Una revisión histórica (notas de los cambios hechos a lo largo del tiempo)

Qué herramientas (compilador, enlazador y otros) son requeridas para compilar el programa

Información adicional que se requiera

Un ejemplo podría ser el siguiente:

```
/*************************************************************************
Programa:           Ejercicio 2.2
Archivo:            ejercicio2.2.cpp
Función:            main (listado completo del programa)
Descripción:        Muestra la palabra Hola en la pantalla
Autor:              Ramón Medina
Ambiente:           Dev-C++ versión 4, Windows XP
Notas:              Programa de ejemplo
Revisiones:         1.00    22/07/2007      Primera versión
                    1.01    25/07/2007      Cambio de Hola por HOLA
*************************************************************************/
```

Es muy importante mantener los comentarios al día. Un problema frecuente es que son ignorados luego de su creación inicial, y con el paso del tiempo, se desactualizan. Cuando son apropiadamente mantenidos, pueden ser de valor incalculable para guiar a través del programa.

Vale la pena mencionar, que los comentarios líneas cuya función se obvia, son completamente inútiles y pueden llegar a ser hasta contraproducentes, en la medida que el programador altere el código y no actualice los comentarios. Sin embargo, lo que puede ser claro para un programador, puede resultar oscuro para otro.

Funciones

Aunque *main*() es una función, es una inusual. Las funciones son llamadas (o invocadas) durante la ejecución de un programa. Un programa es ejecutado línea por línea, en el orden en que estas aparecen en el código fuente, hasta que se encuentra una función. En este momento, la ejecución del programa 'salta' a la función, y cuando esta finaliza, el control es retornado a la instrucción que sigue a la que hizo la invocación. El código a continuación muestra la invocación de una función dentro de un programa.

```
1:  #include <iostream.h>
2:  #include <stdlib.h>
3:
4:  // Nombre: FunciondeDemostracion
5:  // Descripción: Muestra un mensaje por pantalla
6:  void FunciondeDemostracion()
7:  {
8:      cout << "Dentro de la funcion de demostracion\n";
9:  }
10:
11: // Nombre: main
12: // Descripción: muestra un mensaje,
13: // invoca a la FunciondeDemostración
14: // y muestra otro mensaje
15: int main()
16: {
17:     cout << "Dentro de la funcion main\n";
18:     FunciondeDemostracion();
19:     cout << "De regreso en la funcion main\n";
20:
21:     system ("PAUSE");
21:     return 0;
22: }
```

La función FunciondeDemostracion está definida de las líneas 6 a la 9. Cuando es invocada, muestra un mensaje por pantalla y regresa. En la línea 15 está el inicio del programa. En la línea 17, la función muestra un mensaje. Luego, en la línea 18 invoca a la función FunciondeDemostracion, ocasionando que el código dentro de ella se ejecute. La línea 8 en la FunciondeDemostracion muestra un mensaje por pantalla y regresa el control a main(), ejecutándose entonces la línea 19 donde se muestra otro mensaje.

Uso de Funciones

Las funciones pueden retornar un valor o void (no retornar ningún valor). Una función que suma dos enteros, podría retornar la suma de ellos, por lo que debería ser definida para que devuelva un valor entero. Una función que sólo muestre un mensaje, no tiene nada que devolver, por lo cual es declarado para retornar void.

Las funciones consisten de un encabezado y un cuerpo. El encabezado consiste en el tipo de retorno, el nombre de la función y los parámetros. Los parámetros de función, permite que le sean suministrados valores con los cuales operar. Una función que sume dos números, debería recibir como parámetros, los dos números a sumar. La sentencia a continuación muestra un encabezado típico:

```
int suma (int a, int b)
```

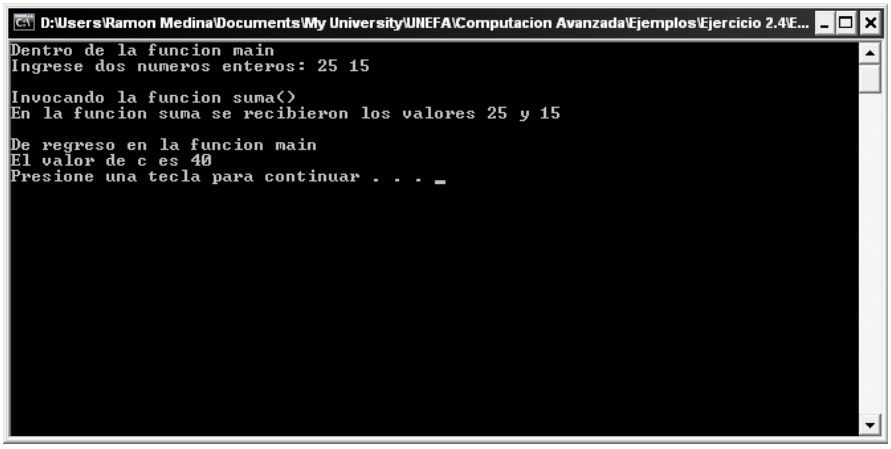
Un parámetro es una declaración de qué tipo de parámetro será proporcionado a la función. El valor actual que es transferido a la función se llama argumento. Muchos programadores usan ambos términos, parámetro y argumento, como sinónimos.

El cuerpo de la función consta de una llave abierta, cero o más sentencias y una llave cerrada. Las sentencias son las que llevan a cabo el trabajo de la función. Una función puede devolver un valor, usando una sentencia return. Esta

sentencia también hace que la función termine su ejecución. Si no se incluye una sentencia return en la función, ésta automáticamente retornará void al final de la función. El valor devuelto debe ser del tipo declarado en el encabezado de la función.

El listado a continuación demuestra el uso de una función que recibe dos parámetros enteros y devuelve un resultado entero:

```
1:  #include <iostream.h>
2:  #include <stdlib.h>
3:  int suma (int x, int y)
4:  {
5:      cout << "En la funcion suma se recibieron los valores " << x << " y " << y << "\n";
6:      return (x+y);
7:  }
8:  int main()
9:  {
10:     int a, b, c;
11:     cout << "Dentro de la funcion main\n";
12:     cout << "Ingrese dos numeros enteros: ";
13:     cin >> a;
14:     cin >> b;
15:     cout << "\nInvocando la funcion suma()\n";
16:     c= suma(a,b);
17:     cout << "\nDe regreso en la funcion main\n";
18:     cout << "El valor de c es " << c << "\n";
19:     system ("PAUSE");
20:     return 0;
21: }
```



La función suma está definida entre las líneas 3 y 7. Toma dos valores enteros como parámetros y retorna un valor entero. El programa en si comienza en la línea 8. En las líneas 12 al 14, el programa indica al usuario que ingrese dos valores enteros. El usuario escribe cada número, separados entre si por un

espacio, y presiona la tecla Enter. La función main() transfiere los valores ingresados por el usuario como argumentos de la función suma(), en la línea 16.

Ejercicios

1

Escriba un programa que despliegue por pantalla lo siguiente:

2

Descubra y corrija el error en el siguiente programa

```
#include <iostream.h>
#include <stdlib.h>
void main (void)
{
     cout << Aqui hay un error" << endl;
     system("PAUSE");
}
```