

Universidad Rafael Urdaneta
Cátedra: Programación I (Laboratorio)
Prof. Rainier Araujo

Guía de Ejercicios N° 1

Introducción al Compilador

Esta guía es meramente para uso del estudiante para practicar los contenidos impartidos en clase, no posee ninguna ponderación para la calificación final. El contenido de esta guía puede ser discutido en clase y entre los alumnos que cursan la asignatura, cualquier duda o proposición puede ser consultada con el profesor de la asignatura. Estos ejercicios están planteados para realizarse en la Herramienta Borland C++ o BloodShed Dev C++. Si se desea utilizar otra herramienta debe ser consultado con anterioridad con el docente.

Programa

Un programa no es más que una serie de comandos o instrucciones que le proporcionamos a un computador para indicarle como realizar una tarea que deseamos que realice. Sin embargo, actualmente el computador no puede comunicarse fluidamente con el ser humano para captar estas instrucciones a través de una conversación. Por lo cual, para poder comunicarle las instrucciones o comandos de nuestro interés debemos hacerlo siguiendo una serie de reglas sintácticas y semánticas.

Lenguaje de programación

A este conjunto de reglas que nos indican cómo debemos darle instrucciones al computador y nos permiten indicarle los pasos para realizar determinada tarea podemos llamarle, se les da el nombre de lenguaje de programación. Existen muchos lenguajes de programación, cada uno con diferentes capacidades y limitaciones, entre los cuales podemos nombrar: C, C++, PHP, Java, Java Script, Python, Ruby, entre muchos otros. En esta asignatura se trabajara con el lenguaje C++.

¿Porque C++?

C++ es uno de los lenguajes con más aplicación en la actualidad; es un lenguaje maduro con una larga trayectoria lo que hace fácil encontrar recursos, comunidades, ejemplos y demás material de soporte para los contenidos obtenidos en clase. Así mismo, es un lenguaje cuya sintaxis es simple y fácil de entender. Dada su trayectoria C++ ha tomado muchos enfoques y se ha adaptado para funcionar en diferentes paradigmas de programación, por lo cual es un lenguaje ideal para servir como iniciación en el área de desarrollo de programas.

Estructura Basica de un Programa en C++

Todo programa en C++, sin importar su funcionalidad, debe seguir una estructura básica.

- 1. Librerías y recursos.**
- 2. Variables y funciones.**
- 3. Metodo main().**

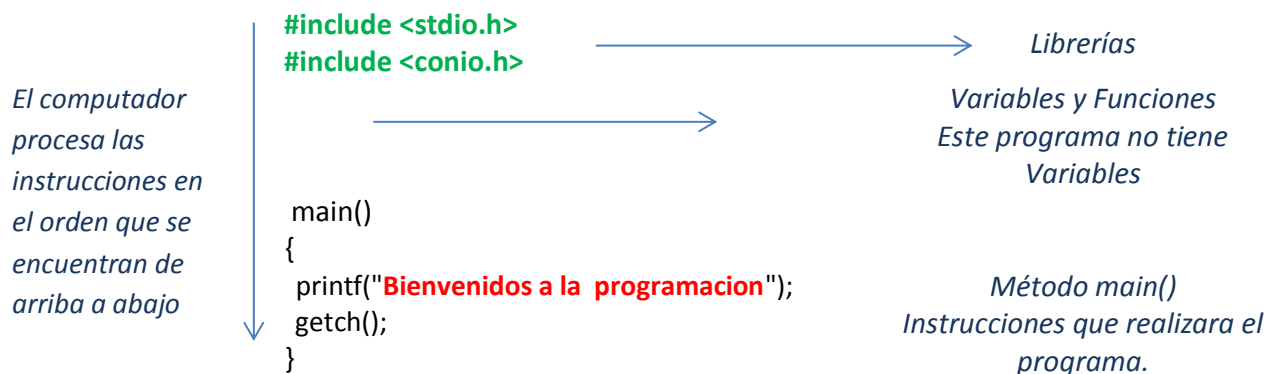
Para entender mejor esta estructura veamos un programa básico en el lenguaje C++ y así observar y conocer mejor sus elementos.

Programa ejemplo C++

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

```
main()
{
    printf("Bienvenidos a la programacion en C++");
    getch();
}
```

Analicemos el ejemplo anteriormente presentado para comprender más a profundidad cada uno de sus elementos y como utilizarlos.



Librerías: una librería es un archivo que contiene un conjunto de funciones, programadas anteriormente, que podemos utilizar en nuestro programa sin necesidad de desarrollarlas completamente de nuevo. En nuestro caso utilizaremos comúnmente las librerías:

`stdio.h`
`conio.h`

Sin embargo, existen otras librerías para funciones más específicas que conoceremos más adelante, como por ejemplo:

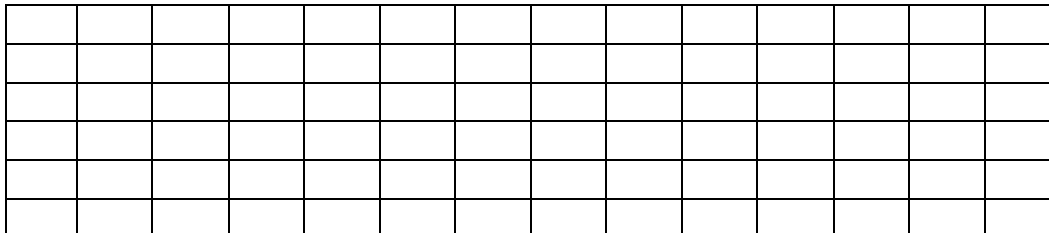
`math.h`
`windows.h`
`stdlib.h`

Veamos ahora el beneficio que obtenemos de usar las librerías en nuestro programa. La función `printf ()` nos permite mostrar un mensaje en pantalla. Cualquier texto que deseemos mostrar podemos hacerlo utilizando esta función. Para poder usar la función `printf ()`, necesitamos la librería `stdio.h`.

Pero si no quisiéramos utilizar una librería, ¿cómo podríamos hacer para lograr este mismo objetivo?. Existen diversas formas, todas igual de extensas y complicadas pero analicemos la más común.

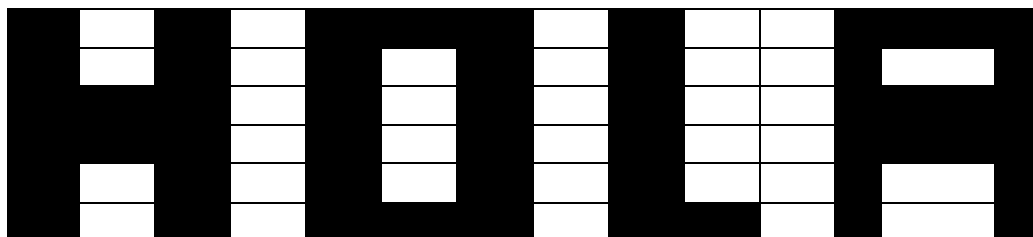
Para mostrar un mensaje en pantalla, se necesita dibujar una serie de letras, números o símbolos que son los que componen dicho mensaje, para hacer que el computador realice esta acción primero debemos entender algunos conceptos básicos.

El monitor, pantalla o display de un computador esta estructurado a manera de una matriz, con cientos de miles de pequeños puntos a los que se les conoce como pixeles.



El coloreado de estos miles de puntos es lo que se utiliza para formar todo tipo de figuras en una pantalla como imágenes, videos, texto, letras y todo elemento que podemos visualizar por el monitor.

Para mostrar nuestro simple mensaje, tendríamos que indicar al computador, que pixeles debe colorear y de qué color para mostrar nuestro mensaje. Uno por uno. La tarea seria extraordinariamente extensa y complicada.



Al utilizar la función printf, solo debemos colocar entre comillas el mensaje que queremos mostrar y la función se encarga de esta tarea por nosotros, convirtiendo una tarea larga y tediosa en una simple instrucción que solo toma momentos completar.

En resumen, las librerías contienen funciones que nos permiten programar de manera mucho más cómoda y rápida por lo cual vale la pena aprender a manejarlas e incluirlas en nuestros programas.

Como incluir una librería al programa

Para poder utilizar las funciones de una librería primero debemos incluirla en el programa que estamos desarrollando. Para incluirla debemos utilizar el comando **#include**. Seguido del nombre de la librería que deseamos incluir entre los signos <nombre de la librería>

Ejemplo:

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

En nuestro programa ejemplo tenemos dos funciones; printf() y getch(), para utilizar estas funciones debemos incluir dos librerías. stdio.h y conio.h ya que printf pertenece a stdio.h y getch pertenece a conio.h.

Variables y funciones

Este programa no contiene variables o funciones, sin embargo, más adelante veremos lo que son las variables y como utilizarlas.

Método main()

El método main es una función que contiene todas las instrucciones que le daremos al computador. Estas instrucciones deben ubicarse entre las llaves { } que siguen al método main.

```
main()
{
    printf("Bienvenidos a la programacion en C++");
    getch();
}
```

Cualquier instrucción que se encuentre fuera de estos símbolos el computador lo tomara como un error ya que no está ubicado correctamente dentro del código. Solo la declaración de variables y la inclusión de librerías pueden estar fuera del método main.

Veamos que instrucciones utilizamos en el programa y cuál es su función:

- printf("Bienvenidos a la programacion en C++");

La función printf como se mencionó anteriormente, permite mostrar un mensaje en pantalla. Siempre que queramos mostrar información usaremos esta función. Es bastante sencilla de utilizar, basta con escribir entre las comillas de la función el mensaje que queremos mostrar y este mensaje será el que aparezca en pantalla.

- printf("Aquí va el mensaje que se quiere mostrar");

Luego sigue la función getch:

- getch();

La función getch sirve para hacer una pausa en el programa, en este caso por ejemplo, al mostrar un mensaje es necesario dar tiempo al usuario para leer dicho mensaje. Esta es la función de getch, el programa hace una pausa hasta que el usuario pulse cualquier tecla para continuar. Si esta función no estuviera en el programa, solo se mostraría el mensaje y terminaría la ejecución sin darle tiempo al usuario de leer la información presentada. En resumen, cuando se necesita una pausa para mostrar información en un programa la función getch nos da una solución ideal.

Dentro del programa pueden utilizarse las funciones printf y getch cuantas veces sea necesario sin problema siempre y cuando estén correctamente ubicadas dentro del código.

Al programar

Al desarrollar programas debemos de ser capaces de asumir dos roles que nos permiten conocer los verdaderos requerimientos de nuestro sistema y evaluar el resultado de manera objetiva. Dichos roles son:

- **Programador:** Es quien desarrolla la aplicación, al asumir el rol de programador debemos siempre buscar el modo más eficiente, con las mejores herramientas y utilizando menos recursos para que nuestro programa funcione de forma óptima.
- **Usuario:** Es quien utiliza el programa, para quien está diseñado, al asumir este rol debemos evaluar la aplicación desde el punto de vista de la comodidad, la organización de la información, los datos solicitados y recibidos.

Ejercicios:

1.- Con los ejercicios realizados en clase como ejemplo desarrolla un programa en el lenguaje C++, que realice las siguientes acciones:

- a) Muestre un mensaje de Bienvenida.
- b) Muestre tus datos personales (nombre, edad, dirección, teléfono)
- c) Cada dato se muestra en una línea diferente (investigar saltos de línea en lenguaje C++).

Compila y ejecuta el programa, trata de reconocer y solucionar las problemáticas que se muestran en el reporte de errores (Si es que los hay).

Puntos extra:

- d) Hacer que el programa entre en reposo por 5 segs entre cada dato (investigar la librería windows.h y el comando sleep())

NOTA: Si el estudiante completa la sección de puntos extra puede enviar el programa via correo electrónico al profesor, si este funciona correctamente el estudiante obtiene una bonificación de puntos en el evaluativo correspondiente a la guía en cuestión.

Cada guía corresponde a un tema. Una vez que se dicta el tema en clase se tiene una semana para enviar los programas para los puntos extra. Una vez transcurrido ese plazo la actividad no otorgara puntos al estudiante.

Para dudas, consultas, puntos extra, información adicional enviar al siguiente correo electrónico:

uru.araujo@gmail.com