



C++ - Módulo 00

Namespace, class, funciones miembro, stdio stream, listas de inicialización, static, const, y montones de cosas básicas

Resumen: Este documento es el cero de los módulos de C++.

Capítulo I

Instrucciones generales

Para los módulos de C++ utilizarás y aprenderás exclusivamente C++98. Tu objetivo es aprender las nociones de la programación orientada a objetos. Sabemos que las últimas versiones de C++ son muy diferentes en muchos aspectos, si quieres volverte un experto en C++ deberás aprender C++ moderno más adelante. Este es el principio de tu largo viaje por C++, es cosa tuya ir más allá después del common core.

- Cualquier función implementada en un header (excepto en el caso de templates), y cualquier header desprotegido, significa un 0 en el ejercicio.
- Todos los output deben ir al standard output, y deben terminar con un salto de línea, salvo que se indique lo contrario.
- Se deben seguir los nombres de archivos impuestos al pie de la letra, así como las clases, funciones y métodos.
- Recuerda: estás programando en C++, no en C. Por lo tanto:
 - Las siguientes funciones están PROHIBIDAS, y su uso será sancionado con un 0, sin preguntas: `*alloc`, `*printf` y `free`.
 - Tienes permitido utilizar básicamente todo de la librería estándar. SIN EMBARGO, sería inteligente utilizar la versión de C++ de las funciones a las que estás acostumbrado en C, en lugar de simplemente seguir utilizando lo que sabes... En realidad, estás aprendiendo un lenguaje nuevo. Y NO, no tienes permitido utilizar STL hasta que debas hacerlo (es decir, el módulo 08). Esto significa que nada de vectors/ lists/maps/etc. O nada que requiera un “include <algorithm>” hasta entonces.
- De hecho, el uso de cualquier función o mecánica explícitamente prohibida será recompensada con un 0, sin preguntas.
- Ten en cuenta que, salvo especificado de otro modo, las palabras de C++ `using namespace` y `friend` están terminantemente prohibidas. Su uso será sancionado con -42, sin preguntas.
- Los archivos asociados con una clase se llamarán siempre `ClassName.hpp` y `ClassName.cpp`, salvo especificado de otro modo.
- Entrega en directorios denominados `ex00/`, `ex01/` ... `exn/`.


- Debes leer los ejemplos con cuidado. Pueden contener requisitos no tan obvios en la descripción del ejercicio.
- Dado que tienes permitido utilizar herramientas de C++ que llevas aprendiendo desde el principio, no tienes permitido utilizar librerías externas. Y antes de que te lo preguntes, esto significa que ningún derivado de C++11, ni Boost o similares se permite.
- Puede que se requiera entregar un importante número de clases. Esto puede parecer tedioso, salvo que sepas instalar scripts en tu editor de texto favorito.
- Lee cuidadosamente los ejercicios POR COMPLETO antes de empezarlos. En serio, hazlo.
- El compilador a usar es `clang++`.
- Tu código debe compilar con las flags: `-Wall -Werror -Wextra`.
- Cada uno de tus `include` debe poder incluirse independientemente del resto. Los `include` deben contener obviamente los `include` de los que dependan.
- Por si te lo preguntas, no se requiere ningún estilo de código durante C++. Puedes utilizar una guía de estilos que te guste, sin limitaciones. Recuerda que si tu evaluador no es capaz de leer tu código, tampoco lo será de evaluarte.
- Algo importante: NO te evaluará un programa, salvo que el subject lo indique explícitamente. Por lo tanto, tienes cierta libertad en cómo hagas los ejercicios. Sin embargo, sé inteligente con los principios de cada ejercicio, y NO seas perezoso, te perderás MUCHO de lo que estos proyectos te pueden ofrecer.
- No es un problema real si tienes archivos adicionales a los que se te solicita, puedes elegir separar el código en más archivos de los que se te piden. Siéntete libre, siempre y cuando el resultado no lo evalúe un programa.
- Aunque el subject de un ejercicio sea corto, merece la pena gastar algo de tiempo para estar absolutamente seguro de que entiendes lo que se espera que entiendas, y que lo has hecho de la mejor forma posible.
- Por Odin, por Thor. Utiliza tu cerebro.

Índice general

I.	Instrucciones generales	1
II.	Ejercicio 00: Megáfono	4
III.	Ejercicio 01: Mi libro de contactos	5
IV.	Ejercicio 02: El trabajo de tus sueños	7

Capítulo II

Ejercicio 00: Megáfono


	Ejercicio: 00
Ejercicio 00: Megáfono	
Directorio de entrega: <i>ex00/</i>	
Archivos de entrega: Makefile , megaphone.cpp	
Funciones prohibidas: Ninguna	

Para asegurarnos de que todo el mundo está despierto, escribe un programa que tenga el siguiente comportamiento:
(Utiliza la forma C++ para interactuar con stdin).

```
$>./megaphone "shhhhh... I think the students are asleep..."
SHHHHH... I THINK THE STUDENTS ARE ASLEEP...
$>./megaphone Damnit " ! " "Sorry students, I thought this thing was off."
DAMNIT ! SORRY STUDENTS, I THOUGHT THIS THING WAS OFF.
$>./megaphone
* LOUD AND UNBEARABLE FEEDBACK NOISE *
$>
```

Capítulo III

Ejercicio 01: Mi libro de contactos

	Ejercicio: 01
Ejercicio 01: Mi libro de contactos	
Directorio de entrega: <i>ex01/</i>	
Archivos de entrega: Makefile , *.cpp , *.{h, hpp}	
Funciones prohibidas: Ninguna	

Bienvenido a los 80 y su tecnología inimaginable. Escribe un programa que se comporte como un inútil increíble libro de contactos. Tómate el tiempo de darle a tu programa un nombre a la altura. Cuando el programa inicie, el usuario tendrá que introducir texto. Debes aceptar los comandos **ADD**, **SEARCH** y **EXIT**. Cualquier otra entrada se descarta.

El programa empieza vacío (sin contactos), no utiliza ningún tipo de reserva dinámica, y no puede guardar más de 8 contactos. Si un noveno se añade, reemplaza el más antiguo.




<http://www.cplusplus.com/reference/string/string/> y por supuesto
<http://www.cplusplus.com/reference/iomanip/>

- Si el comando es **EXIT**:
 - El programa termina y los contactos se pierden para siempre.
- Sin embargo, si el comando es **ADD**:
 - El programa preguntará al usuario, parámetro por parámetro, la información del nuevo contacto hasta llenar todos los parámetros.
 - Un contacto se define por los siguientes parámetros: **nombre** (`first name`), **apellido** (`last name`), **apodo** (`nickname`), **teléfono** (`phone`) y **secreto más oscuro** (`darkest secret`).
 - El PhoneBook debe representarse como la instancia de una clase en tu código que contenga un array de contactos.
 - Un contacto **tiene** que representarse como la instancia de una clase en tu código. Eres libre de diseñar la clase como consideres, pero la evaluación mirará la consistencia de tus elecciones. Mira los vídeos de hoy otra vez si no entiendes lo que esto significa (no quiere decir "utiliza todo").
- En cambio, si el comando es **SEARCH**:
 - El programa mostrará una lista de los contactos existentes en 4 columnas: **índice** (`index`), **nombre** (`first name`), **apellido** (`last name`) y **apodo** (`nickname`).
 - Cada columna debe tener 10 caracteres de ancho, estar alineada a la derecha y separada por el carácter "|". Cualquier salida que supere el ancho de la columna, es truncada y el último carácter imprimible es reemplazado por un punto (".").
 - Por último, el programa pedirá al usuario introducir el índice de la entrada a buscar y mostrará su información de contacto, con los datos de línea en línea. Si la entrada no tiene sentido, establece el comportamiento relevante.
- Si no es ninguno de estos comandos, descarta la entrada.

Cuando un comando se ejecute correctamente, el programa esperará a otro **ADD** o **SEARCH** hasta que se encuentre un **EXIT**.

Capítulo IV

Ejercicio 02: El trabajo de tus sueños

	Ejercicio: 02
Ejercicio 02: El trabajo de tus sueños	
Directorio de entrega: <i>ex02/</i>	
Archivos de entrega: Makefile , Account.cpp , Account.hpp , tests.cpp	
Funciones prohibidas: Ninguna	

Es tu primer día trabajando para **GlobalBanksters United**. Has pasado todas las pruebas técnicas para el equipo de desarrollo gracias a unos trucos de **Microsoft Office** que un amigo te enseñó. La realidad: sabes que lo que les voló la cabeza fue tu increíble manejo de **Adobe Reader**, algo para lo que poca gente está capacitada y que dejó fuera de la competición a tus adversarios.

Da igual cómo, lo has conseguido y tu superior te ha dado la primera tarea: tienes que programar un archivo fuente que falta dado que algo fue mal. **Account.cpp** es el archivo, quizá utilizar el USB en lugar de git para compartir archivos no fue la mejor de las ideas...

Llegado este punto, lo correcto sería dejar ese trabajo; aunque por el bien del ejercicio te vas a quedar.

El archivo **Account.hpp** está presente y una rápida compilación de **tests.cpp** confirma que **Account.cpp** no está presente. También hay un registro que parece contener un resultado equivalente.

Por lo tanto, tienes que crear un archivo **Account.cpp** y desplegar tus habilidades con unas líneas de puro **C++** que, después de unos cuantos fallos de compilación, compilará y pasará los tests con un resultado perfecto... Bueno, salvo por las marcas temporales. Has salvado el día.