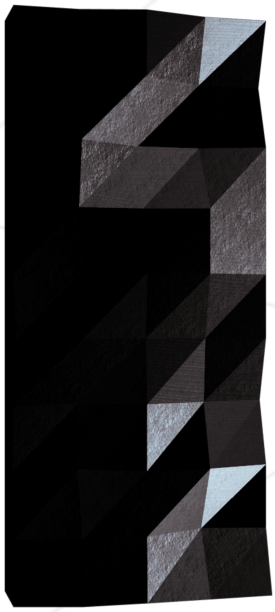

C++ - Módulo 00

Namespace, class, member functions, stdio stream, initialization lists, static, const y muchas más cosas básicas

Resumen: Este documento contiene la evaluación del módulo 00 de la piscina C++ de 42.

Capítulo I

Reglas Generales

- La declaración de una función en un header (excepto para los templates) o la inclusión de un header no protegido conllevará un 0 en el ejercicio.
- Salvo que se indique lo contrario, cualquier salida se mostrará en stdout y terminará con un newline.
- Los nombres de ficheros impuestos deben seguirse escrupulosamente, así como los nombres de clase, de función y de método.
- Recordatorio : ahora está codificando en `C++`, no en `C`. Por eso :
  - Las funciones siguientes están **PROHIBIDAS**, y su uso conllevará un 0: `*alloc`, `*printf` et `free`
  - Puede utilizar prácticamente toda la librería estándar. NO OBSTANTE, sería más inteligente intentar usar la versión para C++ que a lo que ya está acostumbrado en C, para no basarse en lo que ya ha asimilado. Y no está autorizado a utilizar la STL hasta que le llegue el momento de trabajar con ella (módulo 08). Eso significa que hasta entonces no se puede utilizar Vector/List/Map/etc... ni nada similar que requiera un include <algorithm>.
- El uso de una función o de una mecánica explícitamente prohibida será sancionado con un `0`
- Tenga también en cuenta que, a menos que se autorice de manera expresa, las palabras clave `using namespace` y `friend` están prohibidas. Su uso será castigado con un `0`.
- Los ficheros asociados a una clase se llamarán siempre `ClassName.cpp` y `ClassName.hpp`, a menos que se indique otra cosa.
- Tiene que leer los ejemplos en detalle. Pueden contener prerrequisitos no indicados en las instrucciones.
- No está permitido el uso de librerías externas, de las que forman parte `C++11`, `Boost`, ni ninguna de las herramientas que ese amigo suyo que es un figura le ha recomendado.
- Probablemente tenga que entregar muchos ficheros de clase, lo que le va a parecer repetitivo hasta que aprenda a hacer un script con su editor de código favorito.

- Lea cada ejercicio en su totalidad antes de empezar a resolverlo.
- El compilador es `clang++`
- Se compilará su código con los flags `-Wall -Wextra -Werror`
- Se debe poder incluir cada include con independencia de los demás include. Por lo tanto, un include debe incluir todas sus dependencias.
- No está obligado a respetar ninguna norma en `C++`. Puede utilizar el estilo que prefiera. Ahora bien, un código ilegible es un código que no se puede calificar.
- Importante: no va a ser calificado por un programa (a menos que el enunciado especifique lo contrario). Eso quiere decir que dispone cierto grado de libertad en el método que elija para resolver sus ejercicios.
- Tenga cuidado con las obligaciones, y no sea zángano; podría dejar escapar mucho de lo que los ejercicios le ofrecen.
- Si tiene ficheros adicionales, no es un problema. Puede decidir separar el código de lo que se le pide en varios ficheros, siempre que no haya moulinette.
- Aun cuando un enunciado sea corto, merece la pena dedicarle algo de tiempo, para asegurarse de que comprende bien lo que se espera de usted, y de que lo ha hecho de la mejor manera posible.

Índice general

Capítulo II

Ejercicio 00: Megaphone

	Ejercicio : 00
Ejercicio 00: Megaphone	
Directorio de entrega : $ex00/$	
Ficheros a entregar : `Makefile, megaphone.cpp`	
Funciones prohibidas : `Ninguna`	

Para asegurarnos de que esté todo el mundo bien despierto, escriba un programa que se comporte de la siguiente forma:

```
$>./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: My Awesome PhoneBook

	Ejercicio : 01
Ejercicio 01: My Awesome PhoneBook	
Directorio de entrega : $ex01/$	
Ficheros a entregar : `Makefile, *.cpp, *.{h, hpp}`	
Funciones prohibidas : `Ninguna`	

¡Bienvenido a los años 80 y a su increíble tecnología! Escriba un programa increíble que se comporte como un listín telefónico `pésimo`.
Tómese su tiempo para darle a su ejecutable un nombre apropiado. Al iniciarse, el programa solicita el input del usuario: tendrá que aceptar el comando `ADD`, el comando `SEARCH` o el comando `EXIT`. Cualquier otro tipo de entrada será suprimida.

El programa comienza vacío (sin contactos), no utiliza reserva de memoria dinámica y no puede almacenar más de 8 contactos. Si se añade un noveno contacto, defina un comportamiento apropiado.

`http://www.cplusplus.com/reference/string/string/ y, por supuesto, http://www.cplusplus.com/reference/iomanip/`

- Si el comando es `EXIT`:
  - El programa se cierra y se pierden los contactos para siempre
- Si no, si el comando es `ADD`:
  - El programa pedirá al usuario que introduzca nuevos datos de contacto campo a campo, hasta que estén completos todos los campos.
  - Un contacto incluye los siguientes campos: `first name`, `last name`, `nickname`, `login`, `postal address`, `email address`, `phone number`, `birthday date`, `favorite meal`, `underwear color` y `darkest secret`.
  - En su código, un contacto DEBE estar representado como una instancia de clase. Puede crear la clase como le apetezca. No obstante, en la evaluación se comprobará la coherencia de sus decisiones. Si no entiende lo que le quiero decir, vea los vídeos de nuevo (y antes de que me lo pregunte, no me refiero a que lo "utilice todo").
- Por último, si el comando es `SEARCH`:
  - El programa mostrará a 4 columnas una lista de contactos disponibles y que no estén vacíos: `índice`, `nombres`, `apellidos` y `seudónimo`.
  - Las columnas deberán tener un ancho de 10 caracteres con el contenido alineado a la derecha, e irán separadas con el carácter `'|'`. Cualquier salida más larga que el ancho de las columnas será cortada y el último carácter que se pueda mostrar será remplazado por un punto (`'.'`).
  - Después, el programa solicitará de nuevo el índice de la entrada deseada y mostrará la información del contacto, un campo por línea. Si la entrada no tiene ningún sentido, defina un comportamiento apropiado.
- De lo contrario, se ignora el input.

Si se ejecuta correctamente un comando, el programa quedará a la espera de otro comando `ADD` o `SEARCH` hasta que llegue un comando `EXIT`.

Capítulo IV

Ejercicio 02: The Job Of Your Dreams

	Ejercicio : 02
Ejercicio 02: The Job Of Your Dreams	
Directorio de entrega : $ex02/$	
Ficheros a entregar : `Account.class.cpp`	
Funciones prohibidas : `Ninguna`	

`Este ejercicio no puntúa, pero resulta interesante para su piscina.`
`No está obligado a hacerlo.`

Es su primer día de trabajo en `GlobalBanksters United`. Ha superado con éxito las pruebas de selección para el equipo de desarrollo gracias a algunos trucos sobre `Microsoft Office` que un amigo suyo le enseñó. Pero sabe que lo que realmente le impresionó al reclutador fue la rapidez con la que instaló `Adobe Reader`. Eso le ha dado la ventaja suficiente para imponerse ante sus adversarios.

En cualquier caso, lo ha conseguido y el jefe le ha asignado su primera tarea. Le ha explicado que alguien ha perdido el código fuente que gestiona las cuentas de los clientes del banco. El jefe ha decidido volver a compilar y reiniciar el programa para recuperar el archivo. A su predecesor lo despidieron por intentar explicarle cómo tenía que hacer su trabajo: le dijo que “así no es como funciona”, “¡has suprimido el archivo, imbécil!” y “teníamos que haber utilizado Git como te dije”. Qué tipo más arrogante, ¿verdad?

Después de cuarenta minutos de quejas sobre la falta de experiencia de su predecesor, el jefe le pide que escriba para mañana el archivo fuente que falta. Le hubiese encantado hacerlo él mismo, pero ya sabe, tiene que ocuparse de sus tareas de jefe. Así que le envía el archivo `Account.class.hpp` por e-mail, protegido con una clave `gpg`, y la clave privada.

“La seguridad es primordial, los hackers pueden atacar por cualquier lado”, concluye el jefe antes de que usted se vaya de su despacho.

Incómodo por el traje y sudando a chorros, pasa delante de la impresora y la fotocopiadora mientras recorre el largo pasillo que le lleva al espacio abierto donde se estña el sitio que le han asignado. Se encuentra en la zona central y todo el mundo puede ver su pantalla. También se fija en el cartelón colgado en la pared que indica de forma jovial y colorida: "¡Los cascos impiden el espíritu de equipo! ¡El silencio mejora el espíritu de equipo!". El día se le va a hacer muy largo.

Ya sentado delante de su mesa, puede ver en la pantalla la ventana de inicio de sesión de `Windows XP`. A lo largo de la mañana nadie le ha dado su nombre de usuario y contraseña. Le pregunta amablemente a su vecino que a quién hay que pedírselos. Su respuesta “Para las preguntas, abrimos un tique. ¡Es obvio, todo el mundo lo sabe!” no le ayuda mucho. Además, ya empieza a sospechar que mañana le dirán que también es el responsable de los tiques, dado que al parecer es el único empleado IT del edificio. Como no tiene nada que perder, prueba `admin/admin` en la ventana de inicio de sesión. El ordenador se desbloquea y al cabo de un minuto aparecen los iconos y puede utilizar su ordenador.

Después dedica unas horas a entender cómo funciona todo. Se da cuenta de que con `PuTTy` puede conectarse por `ssh`, como root y sin contraseña a un servidor ubicado en algún sitio. El servidor es un antiguo servidor Ubuntu y parece que ejecuta la mayoría de los servicios de la empresa. Consigue crear su cuenta con una identificación apropiada, para dejar de conectarse como root, y bloquea el ssh root. También se da cuenta de que no dispone de cuenta de correo electrónico y que así nunca va a recibir el e-mail con el archivo adjunto de su jefe.

Por fin, gracias a su `bash-fu`, consigue localizar los códigos fuente del proyecto que tiene que arreglar en un directorio que se llama `"test2_REAL_ONE_DONT_DELETE/"`. El directorio contiene algunos archivos escritos entre 1989 y 1992 por un tal Brad MacLane. El archivo `Account.class.hpp` se encuentra ahí y una compilación rápida le confirma que falta el archivo `Account.class.cpp`. También hay un archivo con algunas pruebas y un antiguo logfile, que parece que contiene la salida correspondiente.

A continuación, escribe unas 150 líneas de `C ++` geniales y, tras algunas compilaciones fallidas, su programa compila y supera las pruebas con una salida perfecta, salvo los timestamps. ¡Menudo crack! Pero sus problemas vuelven a la carga cuando oye a su jefe gritar por el pasillo: “¡¿Quién se ha cargado el servidor de producción?! ¡Ya no me puedo conectar!”

Tarea Extra: Añada un atributo a la clase que cuenta el número de llamadas de "int checkAmount(void) const;". Hágalo sin realizar ningún cambio en el prototipo de esta función. Necesita una contraseña nueva y esta es la situación ideal para descubrirla. Evidentemente, esa contraseña no se encuentra en los vídeos.