

(https://profile.intra.42.fr)

SCALE FOR PROJECT CPP MODULE 00 (/PROJECTS/CPP-MODULE-00)

You should evaluate 1 student in this team



Git repository

git@vogosphere-v2.42madrid.com:vogosphere/intra-uuid-4cc7bb0a-(

Introduction

- Evalúa solo el trabajo en el repositorio Git del estudiante o grupo evaluado.
 - Asegúrate de verificar que el repositorio Git pertenece al estudiante evaluado. Asegúrate de que el trabajo que vas a evaluar es el correcto al proyecto evaluado. Asegúrate de que "git clone" se utiliza en un directorio vacío.
 - Comprueba que no existen alias para confundirte y hacerte evaluar contenido que no deberías evaluar.
 - Para evitar sorpresas, tanto el estudiante evaluado como el evaluador deberéis comprobar los scripts utilizados para el proceso de evaluación.
 - Si al evaluar no has completado todavía el proyecto, deberás leer cuidadosamente el enunciado para saber qué se solicita antes de empezar la defensa.
 - Si algo va mal a lo largo del proceso de evaluación, asegúrate de utilizar la flag correspondiente (repositorio vacío, programa errático, fallos de norma, trampas, etc). En cualquiera de estos casos, la evaluación termina y la nota final es un 0 (-42 en caso de trampas). Sin embargo, y salvo en caso de trampa, te recomendamos debatir el trabajo entregado (aunque no esté terminado) para encontrar posibles fallos y evitar que se repitan a futuro.
 - Recuerda que, a lo largo de la evaluación, no se tolerarán segfaults o fallos del programa (por ejemplo, salidas inesperadas, prematuras, descontroladas, etc). En cualquiera de estos casos, la nota final es 0. Utiliza para ello la flag apropiada.
- Nunca deberás editar un archivo, salvo el de configuración si existe.
- En caso de querer editar uno, tanto el estudiante evaluado como el evaluador deberéis estar de acuerdo.

- Deberás verificar la ausencia de memory leaks. Toda la memoria reservada en el heap deberá liberarse correctamente antes de que termine el programa. Tienes permitido utilizar las herramientas disponibles en el equipo, algunos ejemplos son leaks, valgrind, o e_fence. En caso de diagnosticar memory leaks, utiliza la flag apropiada.

Disclaimer

Por favor, sigue las siguientes indicaciones:

- Sé educado, cordial, respetuoso y constructivo a lo largo de toda la evaluación. El bienestar de la comunidad depende de ello.
- Identifica con la persona o grupo evaluado los posibles fallos que pueda presentar el trabajo entregado. Tómate el tiempo de discutir y debatir los problemas que puedan estar presentes.
- Debes entender que no todo el mundo entiende los enunciados del mismo modo, lo que quiere decir que vuestra visión del proyecto o lo que este pide puede variar. Mantén siempre una mente abierta y evalúa de la forma más honesta posible. El buen funcionamiento de la pedagogía parte de una evaluación correcta.

Guidelines

Debes compilar con clang++, utilizando -Wall -Werror -Wextra.

Como recordatorio, este proyecto es en C++98.

Tanto las funciones miembro como los contenedores de C++11 (y posterior) no se permiten.

Si detectas alguno de los siguientes casos, no evalúes el ejercicio en cuestión:

- Una función está implementada en un header (excepto en una template).
- Un Makefile compila sin flags y/o con algo distinto de clang++.


Si detectas alguno de los siguientes casos, debes evaluar el proyecto en cuestión con la flag de función prohibida:

- El uso de una función de "C" (*alloc, *printf, free).
- El uso de una función no autorizada en el enunciado.
- El uso de "using namespace" o "friend".
- El uso de una librería externa, o de elementos de C++20.

Attachments

 subject.pdf (<https://cdn.intra.42.fr/pdf/pdf/35417/es.subject.pdf>)

 Account.hpp (/uploads/document/document/6184/Account.hpp)

 19920104_091532.log (/uploads/document/document/6185/19920104_091532.log)

 tests.cpp (/uploads/document/document/6186/tests.cpp)

Ejercicio 00: Megáfono

Este ejercicio es un calentamiento que busca entender las bases de los i/o streams de C++.

¿Funciona como debe?

Este ejercicio es un programa to_upper con un comportamiento específico cuando se ejecuta sin parámetros. Acepta el enfoque de C++ (strings/upper).

✓ Yes

✗ No

Ejercicio 01: Mi libro de contactos

Este ejercicio es un primer acercamiento a escribir clases simples y un pequeño programa que las utiliza. Si el ejercicio no es completamente funcional, evalúa lo que pueda ser evaluado.

Gestión de errores

Este ejercicio requiere una gestión de errores simple pero los comportamientos no están definidos en el enunciado, así que eliminar o actualizar un contacto viejo es correcto. Un segfault no.

Rate it from 0 (failed) through 5 (excellent)

5

El comando EXIT

Evalúa el comando EXIT según se pide en el enunciado.

✓ Yes

✗ No

Visibilidad

Los atributos de la clase Contact deben ser privados y la clase debe exponer los accesorios correspondientes.

Asegúrate de que únicamente los elementos utilizables fuera de la clase son públicos y los elementos utilizados exclusivamente dentro de la clase son privados. Los principiantes tienden a hacer todo público, no es lo que queremos.

Rate it from 0 (failed) through 5 (excellent)

5

El Contact y la clase PhoneBook

El código debe incluir una clase Contact, aunque el nombre puede variar según decida el estudiante.

Esta clase debe contener atributos para los distintos campos.

El código debe tener una clase PhoneBook con un array de contactos dentro.

☒ Yes☐ No

Bucle de Read/Eval

El programa debe presentar un bucle read/eval en algún punto que debe: leer el input, evaluarlo, y volver a empezar hasta que el comando EXIT aparezca como entrada.

Esta lectura debe hacerse al estilo C++ (std::cin).

☒ Yes☐ No

El comando ADD

Evalúa el comando ADD tal y como se pide en el enunciado.



Rate it from 0 (failed) through 5 (excellent)

El comando SEARCH

Evalúa el comando SEARCH tal y como se pide en el enunciado. Algún ligero desvío en el formato esperado no es relevante. Esta parte busca el uso de iomanip de C++ y es en lo que debes centrarte.



Rate it from 0 (failed) through 5 (excellent)

Ejercicio 02: El trabajo de tus sueños

Este ejercicio busca extraer información y direcciones de ruido inútil, e insertar código nuevo en un contexto ya existente.

¿Has salvado el día?

Este ejercicio va directo al grano; o Account.cpp funciona o no lo hace. Compara el resultado del programa y el log dado. Cualquier diferencia aparte de las marcas de tiempo significa que algo no está bien con el ejercicio.

☒ Yes☐ No

Ratings

Don't forget to check the flag corresponding to the defense

☒ Ok

☐ Outstanding project

☐ Empty work

☐ No author file

☐ Invalid compilation

☐ Norme

☐ Cheat

☐ Crash

☐ Leaks

☐ Forbidden function

Conclusion

Leave a comment on this evaluation

Finish evaluation

- Privacy policy (<https://signin.intra.42.fr/legal/terms/5>)

Terms of use for video surveillance (<https://signin.intra.42.fr/legal/terms/1>)

Rules of procedure (<https://signin.intra.42.fr/legal/terms/4>)

Declaration on the use of cookies (<https://signin.intra.42.fr/legal/terms/2>)

General term of use of the site (<https://signin.intra.42.fr/legal/terms/6>)

Legal notices (<https://signin.intra.42.fr/legal/terms/3>)
- https://projects.intra.42.fr/scale_teams/4068914/edit
- 5/5