

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

SCALE FOR PROJECT CPP MODULE 06 (/PROJECTS/CPP-MODULE-06)

You should evaluate 1 student in this team



Git repository

`git@vogsphere.42urduliz.com:vogsphere/intra-uuid-415c55ca-7`



Introduction

- Evalúa sólo el trabajo entregado en el repositorio Git del estudiante o grupo.
- Comprueba dos veces que el repositorio Git pertenece al estudiante o grupo evaluado. Asegúrate de que el trabajo entregado es el solicitado por el proyecto y que "git clone" es utilizado en una carpeta vacía.
- Comprueba cuidadosamente que no existan alias maliciosos usados para engañarte y hacerte evaluar trabajo subido a un repositorio no oficial.
- Para evitar sorpresas, comprueba cuidadosamente que tanto el estudiante evaluado como el evaluador han revisado los posibles scripts utilizados durante la entrega para facilitar la evaluación.
- Si el evaluador no ha completado este proyecto todavía, es obligatorio para el estudiante leer el subject entero antes de empezar la defensa.
- Utiliza las flags disponibles en esta evaluación para señalar un repositorio vacío, un programa disfuncional, un fallo de norma, trampas, etc. En estos casos, la evaluación termina y la nota final es 0 (o -42 en caso de trampa). Sin embargo, a excepción de trampa, se recomienda seguir discutiendo el trabajo (incluso si no está terminado) para identificar posibles fallos y evitar repetirlos en el futuro.
- Recuerda que durante la defensa, ningún segfault, ni otros comportamientos

prematuramente, descontrolados o cierres del programa se toleran. En caso contrario, la nota final es 0. Utiliza la flag apropiada.

Nunca debes editar ningún archivo salvo el de configuración si existe.

Si quieres editar un archivo, tómate el tiempo de explicar las razones al estudiante evaluado y asegúrate de que ambos estáis de acuerdo.

- Debes también verificar la ausencia de leaks. Toda la memoria localizada en el heap debe liberarse previamente al final de la ejecución.

Tienes permitido usar distintas herramientas disponibles en el ordenador, tales como leaks, valgrind o e_fence. En caso de leaks, utiliza la flag apropiada.

Disclaimer

Por favor respeta las siguientes normas:

- Sé educado, cordial, respetuoso y constructivo durante el proceso de evaluación. El bienestar de la comunidad depende de ello.

- Identifica con la persona o grupo evaluado, las posibles disfuncionalidades del trabajo. Tómate el tiempo de discutir y debatir los problemas que puedes haber identificado.

- Debes considerar la posibilidad de diferir en el entendimiento de las instrucciones del proyecto y el alcance de la funcionalidad. Mantén siempre una mente abierta y evalúa de forma honesta. La pedagogía sólo es válida si las evaluaciones se toman con seriedad.

Guidelines

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

Como recordatorio, este proyecto utiliza C++98.

Tanto las member functions como los contenedores no están permitidos.

Cualquiera de estos errores implica no puntuar el ejercicio en cuestión:

- Una función está implementada en un header (salvo si es una plantilla)
- Un Makefile compila sin flags y/o con algo distinto a clang++

Cualquiera de estos errores implica utilizar la flag de Forbidden Function:

- Utiliza una función de "C" (*alloc, *printf, free)
- Utiliza una función no permitida en el proyecto
- Utiliza "using namespace" o "friend"
- Utiliza librerías externas, o elementos de C++11 y posterior

Attachments

 subject.pdf (<https://cdn.intra.42.fr/pdf/pdf/27284/es.subject.pdf>)

Ejercicio 00: Conversión escalar

Este ejercicio demuestra el uso de static_cast.

Conversión escalar

¿Funciona el programa como está previsto? ¿Utiliza el estudiante static_cast para convertir valores? Aceptamos el uso de conversiones implícitas solo en conversiones de tipo promotion.

De todas formas, no seas demasiado inflexible respecto al resultado del ejercicio si se respeta la intención.

Fallar este ejercicio no implica no evaluar los siguientes.

☒ Yes

☐ No

Ejercicio 01: Serialización

Este ejercicio demuestra el uso de reinterpret_cast.

Rescribir datos sin procesar

¿Funciona el programa como debe?
reinterpret_cast<> deberá utilizarse dos veces.
Una vez desde data* a uintptr_t.
La segunda de uintptr_t a data*.
La estructura final debe ser utilizable.

☒ Yes

☐ No

Ejercicio 02: Identificar el tipo real

Este ejercicio muestra usos prácticos de dynamic_cast.

Identificar el tipo real

¿Funciona el programa como debe?
Verifica el código. ¿Utilizó el estudiante dynamic_casts para identificar el tipo real?
void identify(Base *p) deberá comprobar si el cast devuelve NULL.
void identify(Base &p) deberá utilizarse con un try-catch para verificar fallos de conversión.
(En caso de que te lo preguntes, no debe estar el header en ningún sitio).

☒ 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>)