

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

Remember that the quality of the defenses, hence the quality of the of the school on the labor market depends on you. The remote defences during the Covid crisis allows more flexibility so you can progress into your curriculum, but also brings more risks of cheat, injustice, laziness, that will harm everyone's skills development. We do count on your maturity and wisdom during these remote defenses for the benefits of the entire community.

SCALE FOR PROJECT CPP MODULE 02 (/PROJECTS/CPP-MODULE-02)

You should evaluate 1 student in this team



Git repository

`git@vogsphere-v2.42madrid.com:vogsphere/intra-uuid-59dfe8a5`



Introduction

Por favor, respete las siguientes reglas:

- Durante el proceso de evaluación, sea educado, cortés, respetuoso y constructivo. El bienestar de la comunidad se basa en ello.
- Identifique con la persona (o el grupo) evaluada los posibles fallos del trabajo. Tómese su tiempo para hablar y debate sobre los problemas encontrados.
- Tenga en cuenta que pueden existir pequeñas diferencias de interpretación entre las instrucciones del proyecto, su alcance y sus funcionalidades. Mantenga la mente abierta y sea lo más justo posible a la hora de calificar. La pedagogía funciona únicamente si se realiza evaluación entre pares de forma seria.

Disclaimer

- Califique únicamente el contenido del directorio git clonado del estudiante o del grupo.
- Compruebe que el directorio git pertenece al estudiante o al grupo, que el proyecto es el correcto y que ha utilizado "git clone" sobre un directorio vacío.
- Preste mucha atención a que no se haya utilizado ningún alias para engañarle y asegúrese

de que está corrigiendo la entrega oficial.

- Para evitar cualquier tipo de sorpresa, compruebe con el evaluado los posibles scripts utilizados para facilitar la evaluación.

- Si como evaluador no ha realizado el proyecto en curso, tendrá que leer todo el enunciado antes de empezar la evaluación.

- Utilice los flags disponibles para notificar una entrega vacía, un programa que no funcione, un error de norma, un problema de trampas... En estos casos, se termina la evaluación y la nota es 0 (o -42, en caso de trampas). No obstante, salvo en caso de trampas, se le anima a seguir comentando el trabajo entregado (aunque esté incompleto) para identificar las causas del fracaso y evitar que se vuelvan a reproducir en el futuro.

- Durante la evaluación, no se permitirá ningún segfault ni ninguna parada del programa (inesperada, prematura o incontrolada). En esos casos, la nota final es 0. Utilice el flag adecuado.

No debería tener que editar ningún archivo, salvo algún archivo de configuración, en caso de que exista. Si desea modificar algún archivo, tendrá que explicar claramente las razones del cambio y estar de acuerdo con el estudiante evaluado antes de hacer nada.

- También le corresponde verificar que no existen fugas de memoria. Cualquier memoria reservada en el heap deberá ser liberada correctamente antes de que finalice la ejecución. Para eso, puede utilizar las diferentes herramientas que se encuentran en el ordenador, como leaks, valgrind o también e_fence. En caso de fuga de memoria, marque el flag apropiado.

Guidelines

Debe compilar con clang++, con -Wall -Wextra -Werror

Le recordamos que se trata de un proyecto en C++98 y que no se puede utilizar ninguna función/container de las versiones posteriores.

Si los utiliza, tendrá un 0.

Estos elementos significan que no debe calificar el ejercicio solicitado:

- Se implementa una función en un header (salvo en los templates)
- Un Makefile compila sin los flags y/o con algo que no sea clang++

Estos elementos significan que debe ponerle el tag "Función Prohibida" al proyecto:

- El uso de una función "C" (*alloc, *printf, free)
- El uso de una función prohibida por el proyecto
- El uso de "using namespace" o de "friend"
- El uso de una librería externa o de las características de las versiones posteriores a C++98
- El uso de contenedores ya existentes, o de funciones ya existentes, para implementar otros contenedores

Attachments

 subject.pdf (<https://cdn.intra.42.fr/pdf/pdf/32244/es.subject.pdf>)

Ejercicio 00: Mi primera clase canónica

Este ejercicio introduce la noción de clases canónicas con un simple ejemplo de aritmética; los números de punto fijo.

Makefile

Debe haber un Makefile que compile utilizando las flags apropiadas.

☒ Yes

☐ No

Accesores

La clase Fixed (o como se llame) debe proporcionar accesores a los valores brutos:

- int getRawBits(void) const;
- void setRawBits(int const raw);

¿Están estas funciones miembro presentes y son funcionales?

☒ Yes

☐ No

Clases canónicas

Una clase canónica debe proporcionar como mínimo:

- A default constructor
- A destructor
- A copy constructor
- An assignation operator

¿Están estos elementos presentes y son funcionales?

☒ Yes

☐ No

Ejercicio 01: Avanzando hacia una clase de punto fijo más útil

El ejercicio 00 ha sido un buen punto de inicio, pero nuestra clase todavía es bastante inútil al ser solo capaz de representar el valor de punto fijo 0.0.

Makefile

Debe haber un Makefile que compile utilizando las flags apropiadas.

☒ Yes☐ No

Constructor de punto flotante

¿Es posible construir una instancia desde un valor de punto flotante?

☒ Yes☐ No

Operador <<

¿Hay una sobrecarga del operador << y es funcional?

☒ Yes☐ No

Valor de punto fijo a entero

Una función miembro "int toInt(void) const;" que convierte el valor de punto fijo a un entero debe estar presente. ¿Es funcional?

☒ Yes☐ No

Valores de punto fijo a valores de punto flotante

Una función miembro "float toFloat(void) const;" que convierte un valor de punto fijo a un valor flotante debe estar presente. ¿Es funcional?

☒ Yes☐ No

Constructor de enteros

¿Es posible construir una instancia desde un entero?

☒ Yes☐ No

Ejercicio 02: Ahora estamos hablando

Este ejercicio añade comparación y elementos aritméticos a la clase.

Makefile

Hay un Makefile que compila utilizando las flags pertinentes.

☒ Yes☐ No

División

La clase debe disponer de un operador de división. ¿Está presente y es funcional?

☒ Yes

☐ No

Ejercicio 03: BSP

Este ejercicio debería permitirte darte cuenta de lo fácil que es implementar algoritmos complejos una vez los básicos funcionan como deben.

Makefile

Hay un Makefile que compila utilizando las flags pertinentes.

☒ Yes

☐ No

La clase Point

Hay una clase Point que tiene dos atributos (x, y) de tipo Fixed const. Tiene también un constructor que acepta dos flotantes e inicializa x, y con esos valores.

☒ Yes

☐ No

Función bsp

Hay una función bsp cuyo prototipo es: `[bool bsp(Point const a, Point const b, Point const c, Point const point);]`. La función devuelve true si el punto 'point' está dentro del triángulo descrito por los vértices a, b y c. De lo contrario devuelve false.

☒ Yes

☐ No

Main y tests



Debe haber un main para probar que la función bsp funciona como se espera. Ejecuta diversas pruebas para asegurarte de que el valor de retorno es correcto.

☒ Yes

☐ No

Ratings

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

 Ok Empty work Incomplete work Invalid compilation Cheat Crash 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>)