

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

SCALE FOR PROJECT CPP MODULE 07 (/PROJECTS/CPP-MODULE-07)

You should evaluate 1 student in this team



Git repository

git@vogsphere-v2.42madrid.com:vogsphere/intra-uuid-ef7f2138



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

prematurados, 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++20

Attachments

 subject.pdf (<https://cdn.intra.42.fr/pdf/pdf/35462/es.subject.pdf>)

 main.cpp (/uploads/document/document/6192/main.cpp)

Ejercicio 00: Algunas funciones

En este ejercicio, el estudiante deberá escribir 3 simples templates de funciones; swap, min y max.

Tipos sencillos

Mira el subject para ver el resultado esperado con elementos simples, por ejemplo int.

 Yes

 No

Tipos complejos

¿Es correcto el funcionamiento de la función para tipos complejos?

```
class
Awesome {
public:
Awesome( int n ) : _n( n ) {}
bool operator==( Awesome const & rhs ) const { return (this->_n == rhs._n); }
bool operator!=( Awesome const & rhs ) const { return (this->_n != rhs._n); }
bool operator>( Awesome const & rhs ) const { return (this->_n > rhs._n); }
bool operator<( Awesome const & rhs ) const { return (this->_n < rhs._n); }
bool operator>=( Awesome const & rhs ) const { return (this->_n >= rhs._n); }
bool operator<=( Awesome const & rhs ) const { return (this->_n <= rhs._n); }
private:
int _n;
};
```

 Yes

 No

Ejercicio 01: Iter

Este ejercicio tiene como objetivo escribir una función genérica de iteración a través de arrays.

¿¿¿Funciona???

Prueba el siguiente código con el iterador del evaluado:

```
class Awesome
{
public:
```

```
Awesome( void ) : _n( 42 ) { return; }
int get( void ) const { return this->_n; }
private:
int _n;
};
std::ostream & operator<<( std::ostream & o, Awesome const & rhs ) { o << rhs.get(); return o; }
template< typename T >
void print( T const & x ) { std::cout << x << std::endl; return; }
int main() {
int tab[] = { 0, 1, 2, 3, 4 }; // <--- Qué sentido tiene no poder escribir int[] tab. No sería más simple??
Awesome tab2[5];
```

```
iter( tab, 5, print );
iter( tab2, 5, print );
```

```
return 0;
}
```

Si todo ha ido bien, debería mostrar:

If everything went well, it should display:

```
0
1
2
3
4
42
42
42
42
42
```

☒ Yes

☐ No

Ejercicio 02: Array

En este ejercicio, el estudiante debe escribir una template de clase que se comporte como un array. Si la aloca^{ción} de memoria interna del array actual no viene del uso de new[], no puntués este ejercicio. Pide al estudiante que demuestre el funcionamiento de su template con arrays de elementos simples y complejos antes de validar este ejercicio.

Constructores

¿Es posible crear un array vacío y un array de un tamaño específico?

☒ Yes

☐ No

Acceso

Los elementos deben ser accesibles para lectura y escritura a través del `operator[]` (o sólo para leer si la instancia es constante). Acceder a un elemento fuera de los límites debe lanzar una `std::exception`.

✓ 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>)