

(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.42quebec.com:vogsphere/intra-uuid-7600129a-af` 

Introduction

Merci de respecter les règles suivantes:

- Restez polis, courtois, respectueux et constructifs pendant le processus d'évaluation. Le bien-être de la communauté repose là-dessus.
- Identifiez avec la personne évaluée ou le groupe évalué les éventuels dysfonctionnements de son travail. Prenez le temps d'en discuter et débattre des problèmes identifiés.
- Vous devez prendre en compte qu'il peut y avoir de légères différences d'interprétation entre les instructions du projet, son scope et ses fonctionnalités. Gardez un esprit ouvert et notez de la manière la plus honnête possible. La pédagogie n'est valide que si la peer-évaluation est faite sérieusement.

Guidelines

- Ne notez que ce qui est contenu dans le dépôt Git cloné de l'étudiant(e) ou du groupe.
- Vérifiez que le dépôt Git appartient bien à l'étudiant(e) ou au groupe, que le projet est bien celui attendu, et que "git clone" est utilisé dans un dossier vide.
- Vérifiez scrupuleusement qu'aucun alias n'a été utilisé pour vous tromper et assurez-vous que vous évaluez bien le rendu officiel.
- Afin d'éviter toute surprise, vérifiez avec l'étudiant(e) ou le groupe les

potentiels scripts utilisés pour faciliter l'évaluation (par exemple, des scripts de tests ou d'automatisation).

- Si vous n'avez pas fait le projet que vous allez évaluer, vous devez lire le sujet en entier avant de commencer l'évaluation.

- Utilisez les flags disponibles pour signaler un rendu vide, un programme ne fonctionnant pas, une erreur de Norme, de la triche... Dans ces situations, l'évaluation est terminée et la note est 0, ou -42 en cas de triche. Cependant, à l'exception des cas de triche, vous êtes encouragé(e)s à continuer la discussion sur le travail rendu, même si ce dernier est incomplet. Ceci afin d'identifier les erreurs à ne pas reproduire dans le futur.

- Si le sujet requiert un fichier de configuration, vous ne devriez jamais avoir à le modifier. Si vous souhaitez éditer un fichier, prenez le temps d'expliquer pourquoi à la personne évaluée et de vous assurer que vous avez son accord.

- Vous devez aussi vérifier l'absence de fuites mémoire. Toute mémoire allouée sur le tas doit être libérée proprement avant la fin de l'exécution du programme.

Vous avez le droit d'utiliser tout outil disponible sur la machine tel que leaks, valgrind ou e_fence. En cas de fuites mémoire, cochez le flag approprié.

Attachments

 [subject.pdf \(https://cdn.intra.42.fr/pdf/pdf/58239/fr.subject.pdf\)](https://cdn.intra.42.fr/pdf/pdf/58239/fr.subject.pdf)

Tests préliminaires

Si un cas de triche est suspecté, la notation et l'évaluation prennent fin immédiatement. Pour le signaler, sélectionnez le flag "Cheat". Faites attention à l'utiliser avec calme, précaution et discernement.

Prérequis

Le code doit compiler avec c++ et les flags -Wall -Wextra -Werror
Pour rappel, ce projet doit suivre le standard C++98. Par conséquent, des fonctions C++ 11 (ou autre standard) et les containers ne sont PAS attendus.

Ne notez pas l'exercice si vous trouvez :

- Une fonction implémentée dans un fichier d'en-tête (sauf pour les fonctions templates).
- Un Makefile compilant sans les flags demandés et/ou avec autre chose que c++.

Sélectionnez le flag "Fonction interdite" (Forbidden function) si vous rencontrez :

- L'utilisation d'une fonction "C" (* alloc, * printf, free).
- L'utilisation d'une fonction interdite dans le projet.
- L'utilisation de "using namespace <ns_name>" ou du mot-clé "friend".
- L'utilisation d'une bibliothèque externe, ou de fonctionnalités propres aux versions postérieures à C++98.

☒ Yes☐ No

Exercice 00 : Conversion scalaire

Cet exercice porte sur l'utilisation de `static_cast`.

Conversion scalaire

Assurez-vous que le programme fonctionne correctement et que l'étudiant(e) a eu recours au `static_cast` pour convertir les valeurs.

Les casts implicites sont autorisés pour les "promotion cast" uniquement.

En tout cas, ne soyez pas trop intransigeant(e) vis-à-vis des résultats si l'esprit de l'exercice est respecté.

Si cet exercice est faux, continuez malgré tout l'évaluation.

☒ Yes☐ No

Exercice 01 : Sérialisation

Cet exercice porte sur l'utilisation de `reinterpret_cast`.

Changer le type de data brutes

Le programme fonctionne-t-il comme prévu ?

"`reinterpret_cast<>`" doit être utilisé deux fois :

- Depuis `data*` vers `uintptr_t`.
- Enfin, depuis `uintptr_t` vers `data*`.

Et la structure de données finale devrait être utilisable.

☒ Yes☐ No

Exercice 02 : Identifiez le véritable type

Cet exercice porte sur l'utilisation de `dynamic_cast`.

Identification de type

Le programme fonctionne-t-il comme prévu ?

Vérifiez le code. Est-ce que l'étudiant(e) a utilisé le `dynamic_cast` pour identifier le type véritable de l'objet ?

`void identify(Base* p)` doit vérifier si le retour du cast est NULL.

`void identify(Base& p)` doit utiliser un try-catch pour vérifier si le cast a échoué.

(Au cas où vous vous poseriez la question, le fichier d'en-tête `<typeinfo>` ne doit apparaître nulle part.)

 Yes


 No

Ratings

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

 Ok

 Outstanding project


 Empty work

 Incomplete work

 Invalid compilation

 Cheat

 Crash

 Concerning situation

 Leaks

 Forbidden function

Conclusion

Leave a comment on this evaluation

Finish evaluation

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

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

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

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

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

Legal notices (<https://profile.intra.42.fr/legal/terms/3>)