

≡ 🔍 rrouille

SCALE FOR PROJECT CPP MODULE 00

You should evaluate 1 student in this team



Git repository

git@vogsphere.42lausanne.ch



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

débattez 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](#) [Account.hpp](#)

[19920104_091532.log](#) [tests.cpp](#)

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 : Megaphone

Cet exercice est un échauffement pour découvrir les I/O en C++.

Fonctionnement

Le but de cet exercice est de développer un `to_upper` ayant un comportement spécifique si lancé sans paramètre.
Il doit être résolu avec une approche C++ (`string/upper`).

Yes

No

Exercice 01 : My Awesome Phonebook

Cet exercice est une première approche pour écrire une classe simple et l'utiliser dans un programme interactif. Si l'exercice ne fonctionne pas parfaitement, notez ce qui peut l'être.

Gestion d'erreur

Il y a un peu de gestion d'erreur à faire dans ce programme mais les comportements ne sont pas précisés dans le sujet. Quitter le programme proprement ou gérer les erreurs est ok. Un segfault ne l'est pas ! :D

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



La commande EXIT

Notez la commande EXIT en vous basant sur ce qui est demandé dans le sujet.

Yes

No

Visibilité

Les attributs de la classe Contact doivent être privés. La classe doit exposer les accesseurs correspondants.
Vérifiez également que tout ce qui n'est utilisé que dans une classe (et pas seulement la classe Contact) est privé et le reste public. Les débutants ont tendance à tout mettre en public. C'est ce que vous devez vérifier ici.

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



La classe Contact et la classe Phonebook

Le code doit inclure une classe Contact (ou autre nom). La classe doit contenir les attributs demandés. Le code doit contenir une classe Phonebook avec un tableau de Contacts à l'intérieur.

Yes

No

Boucle de Read/Eval

Le programme doit proposer une sorte de boucle read/eval : Lecture puis traitement d'entrée, ensuite attente de nouvelle entrée jusqu'à recevoir une commande EXIT. La boucle devrait être faite de manière C++ (std::cin) !

Yes

No

La commande ADD

Notez la commande ADD en vous basant sur ce qui est demandé dans le sujet.

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

La commande SEARCH

Notez la commande SEARCH en vous basant sur ce qui est demandé dans le sujet. Une légère divergence dans le format attendu n'est pas importante. Cette partie vise à utiliser les "iomanips" en C++ et c'est ce sur quoi vous devez vous concentrer.

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

Exercice 02 : Le boulot de vos rêves

Le but de cet exercice est de récupérer les informations cachées au milieu du bruit et d'insérer du code dans un contexte existant.

Avez-vous sauvé le monde ?

Cet exercice est plutôt simple. Soit Account.cpp marche, soit il ne marche

pas. Comparez la sortie du programme avec le fichier de logs fourni.

Toute différence (à l'exception des horodatages ou de l'ordre des destructors) signifie que l'exercice est incorrect.

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](#)

[Terms of use for video surveillance](#)

[Rules of procedure](#)

[Declaration on the use of cookies](#)

[General term of use of the site](#)