

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

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

You should evaluate 1 student in this team



Git repository

git@vogsphere.42kocaeli.com.tr:vogsphere/intra-uuid-d27a5b8



Introduction

Lütfen aşağıdaki kurallara uyunuz:

- Değerlendime sürecinde olabildğince saygılı, kibar ve yapıcı davranmaya özen gösteriniz. Topluluğun saygınlığı bu temeller üzerine bağlıdır.
- Çalışmalarını incelediğiniz arkadaşınızın eksiklerini belirleyin. Bu eksiklikler üzerinde tartışın nasıl çözülebileceği üzerinde fikir yürütün.
- Sizi değerlendiren arkadaşlarınızın sizin çalışmanızı nasıl anladığı konusunda farklılıklar olabileceğini her zaman aklınızda bulundurunuz. Değerlendirme sürecinde her zaman açık fikirli olun ve olabildğince dürüst bir değerlendirme yapın. Sistem başarısı arkadaşlarınızın sizi değerlendirirken ne kadar ciddi olduğuna bağlıdır.

Guidelines

- Sadece grubun veya öğrencinin reposundaki dosyaları notlandırınız.
- GiT reposunun öğrenciye ait olup olmadığını iki kere kontrol ediniz. Yüklenen çalışanın projeye ait olduğundan emin olun. Aynı zamanda "git clone" komutunun boş dosya içerisinde kullanıldığını kontrol ediniz.
- Sizi kandırmak ve başka bir şeyi değerlendirmenizi sağlamak için kötü niyetli bir takma adın kullanılmadığını dikkatlice kontrol edin.
- Sürprizlerden kaçınmak için hem değerlendiren öğrencinin hem de değerlendirilen öğrencinin not vermeyi kolaştırmak için kullanılan scriptleri gözgen geçirdiğinden emin olun.

- Eğer değerlendiren öğrenci projeyi tam anlamıyla bitirmemişse savunma sürecinden önce mutlaka tüm proje dosyasını gözden geçirmiş olmalıdır.

- Boş repo, çalışmayan program, norm hatası, kopya gibi durumları işaret etmek için bu kapsamda kullanılabilir olan flagleri kullanın. Bu durumlarda notlandırma sonlanmış olur ve öğrenci 0 alır (eğer kopya tespit edilmişse -42 olacaktır). Eğer kopya dışında bir hata tespit etmişseniz hatanın nedenini öğrenmek ve ileride aynı hatayı bir daha yapmamak için tartışmaya devam edebilirsiniz.

- Unutmayın savunma sürecinde, segfault olmamalı, program beklenmedik bir biçimde sonlanmamalı, premature ve beklenmedik hatalar olmamalıdır. Eğer bunlar olursa son not 0 olacaktır. Uygun flagleri kullanınız.

Eğer konfigürasyon dosyası yoksa bu dosya dışındaki herhangi bir şeyi değiştirmenize gerek yoktur. Eğer edit yapmak istiyorsanız ikinizin de bu durumda anlaşılmış olmanız gerekmektedir.

- Memory sızıntısı olmadığını doğrulamanız gerekmektedir. Heapde ayrılmış olan bölümler program sonlanmadan önce uygun bir şekilde boşaltılmalıdır. Sızıntılar, valgrind veya savunma gibi bilgisayarda bulunan farklı araçlardan herhangi birini kullanmanıza izin verilir. Bellek sızıntısı olması durumunda uygun bayrağı işaretleyin.

Attachments

 subject.pdf (<https://cdn.intra.42.fr/pdf/pdf/78941/tr.subject.pdf>)

Ön testler

Aldatma şüphesi varsa değerlendirme burada durur. Bunu bildirmek için "Hile" bayrağını kullanın. Bu kararı sakince, akıllıca alın ve lütfen bu düğmeyi dikkatli kullanın.

Önkoşullar

Kodun c++ ve -Wall -Wextra -Werror bayraklarıyla derlenmesi gerekir

Bu projenin C++98 standartını takip etmesi gerektiğini unutmayın.

Bu nedenle, C++11 (ve üstü) işlevleri veya kapsayıcıları beklenmez.

Bunlardan herhangi biri, söz konusu alıştırma not vermemeniz gerektiği anlamına gelir:

- Header dosyasında bir işlev uygulanır (şablon işlevleri hariç).
- Bir Makefile, gerekli bayraklar ve/veya c++'tan başka bir derleyici olmadan derlenir.

Bunlardan herhangi biri, projeyi "Yasak İşlev" ile işaretlemeniz gerektiği anlamına gelir:

- Bir "C" işlevinin kullanılması (*alloc, *printf, free).
- Egzersiz yönergelerinde izin verilmeyen bir işlevin kullanılması.
- "<ns_name> ad alanını kullanma" veya "friend" anahtar sözcüğünün kullanılması.
- Harici bir kitaplığın veya C++98 dışındaki sürümlerdeki özelliklerin kullanılması.

☒ Yes☐ No

Alıştırma 00: Ortodoks Kanonik Formda Birinci Sınıfım

Bu alıştırma, kanonik sınıf kavramını basit bir aritmetik örnekle tanıtır: sabit noktalı sayılar.

Makefile

Uygun Flagle kullanarak derlenen bir Makefile var.

☒ Yes☐ No

Aksesuarlar

Sabit sınıf (veya adı ne olursa olsun), ham değere erişimciler sağlamalıdır:

- `int getRawBits(void) const;`
- `void setRawBits(int const raw);` Bu üye işlevleri mevcut ve işlevsel mi?

☒ Yes☐ No

Kanonik

Kurallı bir sınıf en azından şunları sağlamalıdır:

- Varsayılan bir kurucu
- Bir yıkıcı
- Bir kopya oluşturucu
- Bir kopya atama operatörü Bu unsurlar mevcut ve işlevsel mi?

☒ Yes☐ No

Alıştırma 01: Daha kullanışlı bir sabit noktalı sayı sınıfına doğru

Önceki alıştırma iyi bir başlangıçtı, ancak yalnızca sabit nokta değerini 0.0 olarak gösterebildiğinden sınıf hala oldukça işe yaramazdı.

Makefile

Uygun işaretleri kullanarak derlenen bir Makefile var.

☒ Yes☐ No

Kayan nokta oluřturucu

Kayan nokta deęerinden bir rnek oluřturmak mmkn mdr?

☒ Yes☐ No

<< operatr

<< operatr ařırı yklemeři var mı ve iřlevsel mi?

☒ Yes☐ No

Tamsayı deęerine sabit nokta deęeri

Bir ye iřlevi "int toInt(void) const;" sabit nokta deęerini bir tamsayı deęerine dnřtren mevcut iřlev olmalıdır. Bu iřlevsel mi?

☒ Yes☐ No

Sabit nokta deęerinden kayan nokta deęerine

Sabit noktalı deęeri bir kayan nokta deęerine dnřtren bir \ "float toFloat(void) const;" ye iřlevi mevcut olmalıdır. Bu iřlevsel mi?

☒ Yes☐ No

Tamsayı oluřturucu

Bir tamsayı deęerinden bir rnek oluřturmak mmkn mdr?

☒ Yes☐ No

Alıřtırma 02: řimdi konuřuyoruz

Bu alıřtırma, sınıfa karřılařtırma ve aritmetik zellikler ekler.

Makefile

Uygun iřaretleri kullanarak derlenen bir Makefile var.

☒ Yes☐ No

Karşılaştırma operatörleri

6 karşılaştırma operatörü (>, <, >=, <=, == ve !=) uygulandı ve düzgün çalışıyor mu?

☒ Yes☐ No

Aritmetik operatörler

4 aritmetik işleç (+, -, * ve /) uygulanmış ve düzgün çalışıyor mu?

☒ Yes☐ No

Diğer operatörler

Artım öncesi, artım sonrası, eksiltme öncesi ve eksiltme sonrası operatörler uygulanıyor ve düzgün çalışıyor mu?

☒ Yes☐ No

Statik üye işlevleri aşırı yüklemeleri

Son olarak, min() ve max() statik üye işlevlerinin uygulandığını ve düzgün çalıştığını test edin.

☒ Yes☐ No

Alıştırma 03: BSP

Bu alıştırma, temel bilgiler istendiği gibi çalıştığında karmaşık algoritmaları uygulamanın ne kadar kolay olduğunu anlamanızı sağlamalıdır.

Makefile

Uygun işaretleri kullanarak derlenen bir Makefile var.

☒ Yes☐ No

Nokta Sınıfı

Sabit sabit türünde iki özneliği (x ve y) olan bir Point sınıfı vardır. Ayrıca, iki değişken alan ve bu değerlerle x ve y'yi başlatan bir kurucuya sahiptir.

✓ Yes

✗ No

Function bsp

Prototipi "bool bsp(Point const a, Point const b, Point const c, Point const point)" olan bir bsp() işlevi vardır. Nokta a, b, c köşeleri tarafından tanımlanan üçgenin içindeyse işlev True döndürür. Aksi takdirde False döndürür.

✓ Yes

✗ No

Main and testler

bsp() işlevinin gerektiği gibi çalıştığını test etmek için en azından bir main vardır. Dönüş değerinin doğru olduğundan emin olmak için birkaç test çalıştırın.

✓ 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 evaluationİç Yönerge (<https://profile.intra.42.fr/legal/terms/8>)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>)Terms of use for video surveillance (<https://profile.intra.42.fr/legal/terms/1>)Legal notices (<https://profile.intra.42.fr/legal/terms/3>)