



دانشگاه علم و صنعت ایران

دانشکده مهندسی کامپیوتر

برنامه‌سازی پیشرفته
تمرینهای سری اول

مدرس: سید صالح اعتمادی
طرح تمرین: محمدمهدی جاوید
امیرحسین سماوات
پریا فصاحت

مهلت ارسال:
شنبه ۲۳ اسفند ۹۹

فهرست مطالب

۳	۱	مقدمه
۳	۱.۱	موارد مورد توجه
۳	۲	قواعد نام گذاری
۳	۳	کار در کتاب خانه
۳	۱.۳	Book Class C++
۳	۱.۱.۳	Constructor
۳	۲.۱.۳	Setter , Getter
۳	۲.۳	Library Class
۳	۱.۲.۳	Constructor
۳	۲.۲.۳	Add Book
۳	۳.۲.۳	Setter , Getter
۳	۴.۲.۳	Copy Constructor
۳	۵.۲.۳	Destructor
۵	۳.۳	Member
۵	۱.۳.۳	Constructor

5	Registration	2.3.2
5	Destructor	2.3.2
5	Library Features	4.3.2

۱ مقدمه

۱.۱ موارد مورد توجه

- توجه داشته باشید که برای کسب نمره‌ی قبولی درس کسب حداقل نصف نمره‌ی هر سری تمرین الزامی می‌باشد.
- مهلت ارسال پاسخ تمرین تا ساعت ۲۳:۵۹ روز اعلام شده است. توصیه می‌شود نوشتن تمرین را به روزهای نهایی موکول نکنید.
- همکاری و هم‌فکری شما در حل تمرین مانعی ندارد، اما پاسخ ارسالی هر کس حتماً باید توسط خود او نوشته شده باشد.
- مبنای درس، اعتماد بر پاسخ ارسالی از سوی شماست؛ بنابراین ارسال پاسخ در ریپازیتوری گیت شما به این معناست که پاسخ آن تمرین، توسط شما نوشته شده است. در صورت تقلب یا اثبات عدم نوشتار پاسخ حتی یک سوال از تمرین، برای هر دو طرف تقلب‌گیرنده و تقلب‌دهنده نمره‌ی مردود برای درس در نظر گرفته خواهد شد.
- توجه داشته باشید که پاسخ‌ها و کدهای مربوط به هر مرحله را بایستی تا قبل از پایان زمان مربوط به آن مرحله، در سایت [Azure DevOps](#) (طبق توضیحات کارگاه‌ها و کلاس‌ها) بفرستید. درست کردن `Pull request` و `Complete` کردن `Pull request` و انتقال به شاخه‌ی `master` پس از تکمیل تمرین فراموش نشود!
- پس از پایان مهلت ارسال تا ۲ روز به ازای هر روز تاخیر ۱۰ درصد از نمره مربوط به تمرین کسر خواهد شد و پس از ۲ روز نمره‌ای به تمرین تعلق نخواهد گرفت.

۲ قواعد نام گذاری

قواعد نام‌گذاری تمرین را از جدول ۱ مطالعه کنید.

جدول ۱: قراردادهای نام‌گذاری تمرین

Naming conventions		
Branch	Directory	Pull Request
fb_A1	A1	A1

- * ابتدا دایرکتوری به اسم `Assignments` درست کرده و سپس داخل آن موارد پایین را انجام دهید.
- * در کل یک دایرکتوری به نام `A1` بسازید و داخل آن چهار دایرکتوری به نام های `cs` ، `javascript` ، `python` ، `cpp` داشته باشید و فایل های مربوط به هر زبان را داخل دایرکتوری مربوطه بگذارید.
- * برای مثال اگر تمرین `python` وجود ندارد داخل آن دایرکتوری خالی باشد.
- * `C:/git/AP9902/Assignments/A1/cpp/A1_test.cpp`
- * فایل `yml` قرار داده شده است. پایپلاین را درست کنید.
- * برنج `fb_A1` را حذف نکنید.
- * در حال حاضر تست ها روی `Azure` پاس نمی شوند. در آینده فایل تست درست در اختیاران قرار خواهد گرفت.

۳ کار در کتاب خانه

دیمو که به تازگی برنامه نویسی شروع کرده قصد دارد در این روزهای نزدیک به تعطیلات عید یک پروژه ای شروع کند و آن هم کتابخانه مجازی است. برای همین اول میخواهد اطلاعات کتاب ها را ذخیره کند.

۱.۳ Book Class C++

در ابتدا فایل تست بیلد نمیشود علت آن را با نگاه کردن به کلاس ها بیابید.
هر تستی را که از کامنت در می آورید تابع های مورد نیاز آن را از فایل test_func.hpp نیز از کامنت در بیاورید
* چیزی از فایل های داده شده را پاک نکنید.

۱.۱.۳ Constructor

- ابتدا Default Constructor را پیاده سازی کنید.
- Parameter Constructor را با استفاده از Initializer List پیاده سازی کنید.
- * ورودی های Constructor به صورت Designated Initializers داده شده است.
- بخش Copy Constructor را با this Pointer پیاده سازی کنید.

۲.۱.۳ Setter ، Getter

- برای هر کدام از Data Member ها Getter را پیاده سازی کنید.
- در بخش Setter قیمت ها را برای سود بیشتر تا دو رقم رو به بالا گرد کنید.
- در بخش Setter امتیاز اگر عددی بین ۱ تا ۵ بود امتیاز آن کتاب تغییر کند, در غیر این صورت امتیاز قبلی مد نظر است.

۲.۳ Library Class

دیمو پس از اینکه فهمید آمریکا با کمبود برنامه نویس مواجه شده است از این فرصت استفاده کرد و به چند کتابخانه پیشنهاد همکاری داد و آن ها هم قبول کردند.

۱.۲.۳ Constructor

- ابتدا Default Constructor را پیاده سازی کنید.
- Parameter Constructor را با استفاده از Initializer List پیاده سازی کنید.
- * ورودی های Constructor به صورت Designated Initializers داده شده است.

۲.۲.۳ Add Book

یک کتاب به کتابخانه اضافه میکند.

۳.۲.۳ Setter ، Getter

- برای هر کدام از Data Member ها Getter و Setter مناسب را پیاده سازی کنید.

۴.۲.۳ Copy Constructor

- در این بخش فقط کتاب های کتابخانه مورد نظر کپی شود.

۵.۲.۳ Destructor

- این بخش را به گونه ای پیاده سازی کنید که Memory Leak ندهد.

* این تست در Pipeline CPP Azure به گونه دیگری پیاده سازی می شود, این تست را در موقع Push کردن Comment کنید.

- * در صورت پیاده سازی درست بدون Memory Leak و Pass نشدن آن را چند بار اجرا کنید.

Member ۳.۳

با تمام شدن بخش کتابخانه حال دیمو می‌خواهد بخش عضویت در کتابخانه را پیاده سازی کند. این بخش به او کمک کنید.

Constructor ۱.۳.۳

- ابتدا `Default Constructor` را پیاده سازی کنید.
- `Parameter Constructor` را پیاده سازی کنید.
- برای هر کدام از `Data Member` ها `Getter` را پیاده سازی کنید.
- در بخش `Copy Constructor` فقط موجودی حساب افراد مورد نظر کپی شود.

Registration ۲.۳.۳

- هزینه ثبت نام ۱۰۰۰ تومان است و در صورت عدم وجود مبلغ در کتاب خانه عضو نخواهند شد.
- کتابخانه باید تمام افراد ثبت نام شده را ذخیره کند.

Destructor ۳.۳.۳

- این بخش را به گونه ای پیاده سازی کنید که `Memory Leak` ندهد.
- * برای تست کردن این بخش باید `MembersMemoryLeak` را در `test_func.hpp` از `Comment` در بیاورید.
- * این تست در `Pipeline CPP Azure` به گونه دیگری پیاده سازی می شود، این تست را در موقع `Push` کردن `Comment` کنید.
- * در صورت پیاده سازی درست بدون `Memory Leak` و `Pass` نشدن آن را چند بار اجرا کنید.

Library Features ۴.۳.۳

بعد از اتمام کار های بالا توسط شما و دیمو، از دیمو خواسته شده است که امکانت جدیدی که خودش به ذهنش میرسد به کتاب خانه اضافه کند.

- ابتدا قرار است بر حسب حروف الفبا ثبت نام کنندگان هر کتابخانه را به صورت جدا `Sort` کنید.
- تابع `BorrowBook` را به گونه ای پیاده سازی کنید که $1/4$ قیمت کتاب را به عنوان پیش پرداخت از حسابشان کم کند. در صورت کافی نبودن موجودی کتاب قرض داده نمی شود.
- * هر کتاب دران واحد به یک نفر قرض داده می شود و هر فرد یک کتاب را میتواند قرض بگیرد و مدت زمان قرض دادن کتاب و فردی که کتاب را قرض گرفته در آن ذخیره می شوند
- * تمامی محاسبات تقسیم را از `int` استفاده کنید.
- تابع `DaysPassed` را به گونه ای پیاده سازی کنید که هر کتابی که قرض گرفته شده بود به ازای هر روز $1/10$ قیمت کتاب از موجودی فرد کم می شود.
- * در صورتی که مدت زمان قرض گرفته شده رد شده بود هزینه ی بالا 2 برابر شود.
- * در صورتی که موجودی فرد به 0 رسید فرد را پیدا کرده و اخراج کنید و کتاب موجود خواهد شد و تعداد روز های قرض گرفته شده آن 0 خواهد شد.
- * تمامی محاسبات تقسیم را از `int` استفاده کنید.
- تابع `Find Books` را به گونه ای پیاده سازی کنید که دسته بندی و امتیاز آن را گرفته و کتاب مورد نظر را برگرداند.
- * برای پیدا کردن دسته بندی حروف بزرگ و کوچک بی تاثیر است و حتی اگر بخشی از آن داده شود باید آن را پیدا کند.
- * امتیاز باید بزرگتر مساوی باشد.